Algorithme optimal de décision pour l'équivalence des grammaires simples

Didier Caucal

INRIA

CENTRE DE RENNES
IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 344

ALGORITHME OPTIMAL DE DÉCISION POUR L'ÉQUIVALENCE DES GRAMMAIRES SIMPLES

Didier CAUCAL

Octobre 1984

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 - RENNES CÉDEX
FRANCE
Tél. : (99) 36.20.00
Télex : UNIRISA 95 0473 F

Publication Interne n° 238

Septembre 1984
48 pages

ALGORITHME OPTIMAL DE DECISION POUR L' EQUIVALENCE DES GRAMMAIRES SIMPLES

Didier CAUCAL

Abstract *. We decide the syntactical equivalence for simple grammars by an algorithm increasing the Korenjak-Hopcroft 's one but with a linear and optimal complexity .*

Résumé *. On traite le problème de l' équivalence des grammaires simples par un algorithme renforçant celui de Korenjak-Hopcroft mais de complexité linéaire et minimale .*

1 Introduction

Un problème important dans l' étude des langages algébriques , langages engendrés par des grammaires dites algébriques ou bien langages reconnus par des automates à pile non déterministes [Ha 78] [Ho-Ul 79] , est la décidabilité de l' égalité de ces langages . Cette dernière est indécidable à moins de se restreindre à des classes de langages algébriques . Ce problème reste ouvert en particulier pour la classe (on compare les langages d' une même classe) des langages algébriques déterministes ou de façon similaire pour l' équivalence syntaxique (ou forte) des schémas récursifs monadiques [Fr 77] . Par contre , des algorithmes de décision existent notamment pour la classe des langages simples [Ko-Ho 66] [Wo 73] [Ha-Ha-Ye 79] et pour la classe des langages LL(k) [Ol-Pn 77] [To 83] . Une formulation axiomatique a été donnée [Co 83] .

Informellement , ces algorithmes reposent sur la construction , pour toute paire de langages à comparer , d' un arbre fini , dit de transformation (ou de comparaison) , opérant directement sur les paires de mots des non-terminaux des grammaires génératrices mises préalablement en forme normale de Greibach . Le développement de l' arbre , dont la racine est la paire d' axiomes engendrant les deux langages à comparer , s' effectue en parallèle à l' aide de règles de transformation locales (on applique une des transformations au label du noeud à développer pour obtenir ces descendants immédiats) et valides (un noeud non terminal à un label équivalent si et seulement si tous ces descendants aussi ; avec pour label équivalent , une paire d' axiomes engendrant un même langage) . Un noeud n' est plus développable si son label est soit label d' un noeud déjà développé ou soit d' équivalence trivialement décidable (paire réflexive ou si les langages engendrés se différencient dès une première lettre terminale)

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L.A. 227)
UNIVERSITÉ DE RENNES 1 I.N.S.A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

L' apport de ce papier est de compléter ces algorithmes en adjoignant au développement de tout arbre de transformation une condition plus forte d' arrêt qui est celle de conséquence théorique [Hu 80] c' est à dire un noeud n' est plus développable si en plus son label est conséquence théorique des labels des noeuds précédemment développés. Appliqué à la classe des grammaires simples, on montre que la complexité de l' algorithme obtenu est optimale ou plus exactement, ce que l' on qualifie de critère d' équivalence, que l' algorithme permet la détermination d' une borne minimale, entier dépendant de caractéristiques des deux langages simples déterministes à comparer, tel que les deux langages sont égaux si et seulement si tout facteur gauche, de taille inférieure à cet entier, d' un de ces langages l' est de l' autre. Un tel résultat a été déjà approché de façon combinatoire [Bu 73]. Cette méthode est généralisable à la classe des grammaires LL(k) mais son domaine d' application reste à étudier. Des travaux complémentaires portant sur les grammaires simples ont été présentés [Co 74] [Fr 76].

Ce travail est exposé de façon à ce qu' il n' y ait aucun acquis préalable. Le paragraphe 2 est celui des préliminaires. On introduit toutes les notations et définitions nécessaires. Le paragraphe 3 présente l' algorithme de décision avec quelques exemples et le paragraphe 4 énonce le critère d' équivalence en montrant son optimalité. Le paragraphe 5 prouve la validité de l' algorithme et enfin le paragraphe 6 évalue la complexité de l' algorihme c' est à dire vérifie le critère d' équivalence.

2 Notations et rappels

2.1 Mots et langages

Soient $\mathbb{N}$ l' ensemble des entiers naturels et $\mathbb{N}_+ = \mathbb{N}-\{0\}$

On pose pour $i \in \mathbb{N}$, $[i] = \{1, \ldots, i\}$ avec $[0] = \emptyset$

c.a.d. $\forall\, i \in \mathbb{N}$, $[i]$ = **si** $i=0$ **alors** $\emptyset$ **sinon** $\{i\} \cup [i-1]$ **is**

Soit Σ^* l'ensemble des mots de longueurs finies que l' on peut constituer par concaténation à partir de l' alphabet Σ

c.a.d. $(\Sigma^*, .)$ est le monoïde libre engendré par l' ensemble Σ à l' aide de l'opération (de concaténation) ".". On note $\perp$ le neutre de Σ^* pour ".". $\perp$ est appelé le mot vide.

$\forall\, u \in \Sigma^*$, on désigne par $|u|$ la longueur du mot u et par u(i) le $i^{\text{ème}}$ élément de Σ constituant u.

c.a.d. $\forall\, u \in \Sigma^*$, $|u|$ = **si** $u=\perp$ **alors** 0 **sinon** $1+|v|$ avec ($u = av$ et $a \in \Sigma$) **is**

et $\forall\, i \in \mathbb{N}$,

$$u(i) = \begin{cases} \perp & \text{si } i = 0 \text{ ou } u=\perp \\ a & \text{si } i = 1 \text{ et } \exists\, v \in \Sigma^* \,/\, u = av \text{ et } a \in \Sigma \\ v(i-1) & \text{si } i>1 \text{ et } \exists\, v \in \Sigma^* \,/\, u = av \text{ et } a \in \Sigma \end{cases}$$

On note $\forall\, i \in \mathbb{N}$, $\forall\, u \in \Sigma^*$,

$$u\backslash i = \begin{cases} u(i). \ldots .u(|u|) & \text{si } i \in [\,|u|\,] \\ \perp & \text{sinon} \end{cases}$$ le suffixe du mot u de longueur $|u|-i+1$

On définit, pour $u \in \Sigma^*$, $FG(u) = \{v \in \Sigma^* \,/\, \exists\, w \in \Sigma^* \text{ et } u = vw\}$ l' ensemble des facteurs gauches (ou préfixes) du mot u.

On étend l' opération de concaténation aux langages sur Σ (parties de Σ^*) par :

$\forall\, L, L' \subset \Sigma^*$, $L.L' = \{xy \,/\, x \in L, y \in L'\}$

On pose $\Sigma^+ = \Sigma.\Sigma^*$ c.a.d. $\Sigma^*-\{\perp\}$

et $\forall\, n \geqslant 0$, $\Sigma^{n+1} = \Sigma.\Sigma^n$ l' ensemble des mots de longueur n+1 où $\Sigma^0 = \{\perp\}$

$FG(L) = \bigcup_{u \in L} FG(u)$

On note $P(A)$ l' ensemble des parties d' un ensemble A et $Card(A)$ le nombre de ses éléments

PAPIER RECUPERE ET RECYCLE

2.2 Grammaires algébriques simples et réduites

On note pour R relation binaire

quelconque

$Ker(R) = \{x / (x,y) \in R\}$

$Im(A,R) = \{y / \exists x \in A \text{ et } (x,y) \in R\}$ l' image de l' ensemble A par la relation R

<u>sur</u> un ensemble E quelconque ($R \subset E \times E$)

R_n pour $n \in \mathbb{N}$, défini inductivement par

$R_0 = \{(x,x) / x \in E\}$ noté aussi Id_E qui est la relation d' identité sur E

et $\forall n \geqslant 0$, $R_{n+1} = \{(x,z) \in E \times E / \exists y \in E$ où $(x,y) \in R_n$ et $(y,z) \in R\}$

R^* sa fermeture réflexive et transitive c.a.d. $R^* = \bigcup_{n \geqslant 0} R_n$

$R^{-1} = \{(y,x) / (x,y) \in R\}$ la relation symétrique de R

et si $+$ est une opération binaire interne sur E , on dit que

R est régulière pour $+$ $\Leftrightarrow$ $\forall a \in E$, $\forall (x,y) \in R$ $(x+a, y+a), (a+x, a+y) \in R$

R est stable pour $+$ $\Leftrightarrow$ $\{(x+s, y+t) / (x,y), (s,t) \in R\} \subset R$

R est simplifiable pour $+$ $\Leftrightarrow$

$$\forall (x+x', y+y') \in R \quad \left| \begin{array}{l} \text{si } (x,y) \in R \text{ alors } (x',y') \in R \\ \text{si } (x',y') \in R \text{ alors } (x,y) \in R \end{array} \right.$$

$\vec{R}$ la fermeture de R par régularité pour $+$ c.a.d.

$$\forall x,y \in E,\ x \underset{R}{\rightarrow} y \Leftrightarrow \exists a,b \in E, \exists (x',y') \in R /$$
$$x = a+x'+b \text{ et } y = a+y'+b$$

La relation binaire G est dite grammaire algébrique sur l' alphabet Σ (dit ensemble des terminaux) si par définition

$$G \text{ est finie et } Im(Ker(G),G) \subset (Ker(G) \cup \Sigma)^*$$

On pose $\mathbf{N} = Ker(G)$ dit ensemble des non-terminaux de la grammaire G

G est alors une relation binaire sur $(\Sigma \cup \mathbf{N})^*$, et donc la fermeture de G par régularité pour l' opération de concaténation "." est parfaitement définie

$$\text{i.e. } \forall U,V \in (\Sigma \cup \mathbf{N})^*,\ U \underset{G}{\rightarrow} V \Leftrightarrow \exists X,Y \in (\Sigma \cup \mathbf{N})^*, \exists (A,W) \in G /$$
$$U = XAY \text{ et } V = XWY$$

On introduit

$L(G,U) = \{u \in \Sigma^* / U \overset{*}{\underset{G}{\to}} u\}$ le langage des (mots) terminaux engendré par G à partir de l' axiome $U \in (\Sigma \cup \mathbf{N})^*$ où $\overset{*}{\underset{G}{\to}}$ désigne $(\underset{G}{\to})^*$

c.a.d. $\forall\, U \in (\Sigma \cup \mathbf{N})^*$, $L(G,U) = \mathrm{Im}(\{U\}, \overset{*}{\underset{G}{\to}}) \cap \Sigma^*$

On définit la relation d' équivalence (syntaxique) selon G, notée $\equiv_G$, binaire sur $(\Sigma \cup \mathbf{N})^*$ par :

$\forall\, U,V \in (\Sigma \cup \mathbf{N})^*$, $U \equiv_G V \iff L(G,U) = L(G,V)$

On dit que G, grammaire algébrique sur Σ^*, est

réduite $\iff \forall\, A \in \mathbf{N}$, $L(G,A) \neq \emptyset$

$\perp$-libre $\iff \forall\, A \in \mathbf{N}$, $\perp \notin L(G,A)$

en forme normale de Greibach, notée FNG $\iff \mathrm{Im}(\mathbf{N},G) \subset \Sigma.\mathbf{N}^*$

simple $\iff$ G est en FNG et $\forall A \in \mathbf{N}$, $\forall a \in \Sigma$, $\mathrm{Card}(\mathrm{Im}(\{A\},G) \cap \{a\}.\mathbf{N}^*) \leqslant 1$

On définit $\forall\, U \in (\Sigma \cup \mathbf{N})^*$, $\tau_G(U) = \begin{cases} \inf\{\, |u| \,/\, u \in L(G,U)\} & \text{si } L(G,U) \neq \emptyset \\ \infty & \text{sinon} \end{cases}$

τ_G est un homomorphisme de $((\Sigma \cup \mathbf{N})^*, .)$ dans $(\mathbb{N} \cup \{\infty\}, +)$.

De plus, G est $\perp$-libre $\iff \forall\, A \in \mathbf{N}$, $\tau_G(A) > 0$

G est réduite $\iff \forall\, A \in \mathbf{N}$, $\tau_G(A) < \infty$

Comme en [Co 74], τ_G est appelée la valuation de G

Un langage L sur Σ est dit

algébrique $\iff \exists$ G grammaire algébrique, $\exists\, U \in (\Sigma \cup \mathbf{N})^* / L = L(G,U)$ où $\mathbf{N}$=Ker(G)

(algébrique) simple $\iff$ il existe une grammaire (algébrique) simple qui, à partir d' un axiome donné, engendre L

préfixe $\iff \forall\, u,v \in L$, $u \notin FG(v)$ si $u \neq v$

On vérifiera les propriétés élémentaires suivantes :

toute grammaire en FNG est $\perp$-libre

tout langage algébrique ne contenant pas $\perp$ peut être engendré par une grammaire en FNG, à partir d' une lettre non terminale

tout langage simple est préfixe

pour toute grammaire G, $\equiv_G$ est stable (et régulière) pour "."

pour toute grammaire G simple et réduite, $\equiv_G$ est simplifiable pour "." (cf 5.1)

Comme, la décidabilité de l' égalité des langages simples est identique à celle de l' équivalence (syntaxique) selon toute grammaire simple et que de plus, on sait pour tout langage algébrique décider

s' il est vide ou non , on peut donc se restreindre à étudier $\equiv_G$ pour G grammaire simple et réduite
Dorénavant , on note

G une grammaire algébrique simple et réduite sur Σ avec $\mathbb{N}$ = Ker(G)

a, b, ... , h des éléments de Σ (lettres terminales)

i, j, ... , t des variables représentatives de $\mathbb{N}$

u, v, ... , z des mots de Σ^* ou $\mathbb{N}_+^*$

A, B, ... , T des éléments de $\mathbf{N}$ (lettres non terminales)

U, V, ... , Z des mots de $(\Sigma \cup \mathbf{N})^*$

α, β, ... des mots de $\mathbf{N}^*$

Φ, Ψ des éléments de $\mathbf{N}^* \times \mathbf{N}^*$

2.3 Arbres

Soit A un ensemble , fini ou non , quelconque . Un A-arbre T est une fonction (partielle) de $\mathbb{N}_+^*$ dans A tels que :

Dom(T) = Ker(T) appelé domaine de l' arbre (c.a.d. son support) est clos par préfixe

i.e. si $uv \in$ Dom(T) alors $u \in$ Dom(T)

tous les "frères gauches" d' un noeud (élément du domaine) existent

c.a.d. $\forall\, i \in \mathbb{N}_+ ,\ ui \in \mathrm{Dom}(T) \Rightarrow \forall\, j \in [i-1] ,\ uj \in \mathrm{Dom}(T)$

Si de plus , A est gradué i.e. A est un ensemble de fonctions munies d' une arité

c.a.d. on adjoint à A , ensemble quelconque , une application , notée ρ et nommée arité , de A dans $\mathbb{N}$

alors un A-arbre T vérifie en plus la condition

$\forall\, u \in \mathrm{Dom}(T) , \forall\, i \in \mathbb{N}_+ ,\ ui \in \mathrm{Dom}(T) \Leftrightarrow i \in [\rho(T(u))]$

autrement dit , le nombre de descendants immédiats de tout noeud est égal à l' arité de son label (image par T du noeud)

Un arbre est dit fini si son domaine l' est . On introduit la taille $\| T \|$, et la hauteur d(T) , d' un arbre fini comme étant respectivement la cardinalité , et la longueur maximale des éléments (mots) , du domaine de l' arbre .

On désigne par $T\backslash u$ le sous-arbre de T au noeud u ($u \in$ Dom(T)) défini par :

$$\mathrm{Dom}(T\backslash u) = \{v \in \mathbb{N}_+^* \ / \ uv \in \mathrm{Dom}(T)\}$$

et $\forall\, v \in \mathrm{Dom}(T\backslash u) ,\ (T\backslash u)(v) = T(uv)$

On note $\mathrm{Fl}(T) = \{u \in \mathrm{Dom}(T) / \ u1 \notin \mathrm{Dom}(T)\}$ l' ensemble des noeuds terminaux (feuilles) de l' arbre T

On désigne par $\leqslant_{lex}$ l' ordre lexicographique sur $\mathbb{N}_+^*$ induit par l' ordre naturel $\leqslant$ sur $\mathbb{N}_+$

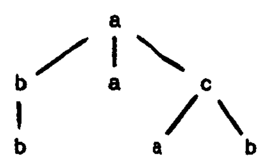
$$\forall\, u,v \in \mathbb{N}_+^* ,\ u \leqslant_{lex} v \Leftrightarrow \left| \begin{array}{l} u = \bot \\ \text{oux } v \neq \bot \ \text{ et } \left| \begin{array}{l} u(1) < v(1) \\ \text{ou } \ u(1)=v(1) \ \text{ et } \ u\backslash 2 \leqslant_{lex} v\backslash 2 \end{array} \right. \end{array} \right.$$

avec oux pour le ou exclusif

On note $<_{lex}$ l' ordre lexicographique strict sur $\mathbb{N}_+^*$

Exemple : Soit $A \supset \{a, b, c\}$ un ensemble non gradué

Notons T le A-arbre de représentation

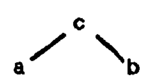


c' est à dire , défini par

$\mathrm{Dom}(T) = \{\perp, 1, 11, 2, 3, 31, 32\}$ classé par ordre lexicographique et de labels associés $\{a, b, b, a, c, a, b\}$

on a $\| T \| = 7$; $d(T) = 2$; $Fl(T) = \{11, 2, 31, 32\}$

Le sous-arbre de T au noeud 3 , $T\backslash 3$, a donc pour représentation



2.4 Transformations

Ce paragraphe n' est qu' une reprise des travaux de [Ha 78] et [Ha-Ha-Ye 79].

G étant une grammaire simple, on peut donc définir

$\forall\ \alpha \in \mathbf{N}$, $\forall$ a $\in \Sigma$, $R(\alpha,a)$ = Si $\exists\ \beta \in \mathbf{N}^* / (\alpha,a\beta) \in G$ **Alors** $a\beta$ **Sinon** $\perp$ **fs**

par commodité, on étend R en posant

$\forall$ a $\in \Sigma$, $R(\perp,a) = a$

On reprend , en les modifiant quelque peu , les deux transformations définies par Harrison , à savoir

a) la transformation parallèle gauche (la plus à gauche), notée T_A , application de $\mathbf{N}^*\times\mathbf{N}^*$ dans $P(\mathbf{N}^*\times\mathbf{N}^*)$, définie pour $\alpha,\beta \in \mathbf{N}^*$ par

$$T_A(\alpha,\beta) = \left|\begin{array}{l} \emptyset \text{ si pour } \{A,B\} = \{\alpha(1),\beta(1)\}\ \exists\ a \in \Sigma\ /\ R(A,a) = \perp \neq R(B,a) \\ \{(R(\alpha(1),a)\backslash 2\ .\ \alpha\backslash 2\ ,\ R(\beta(1),a)\backslash 2\ .\ \beta\backslash 2)\ /\ a\in\Sigma \text{ et } R(\alpha(1),a)\neq\perp\} \text{ sinon} \end{array}\right.$$

Exemple : pour $G = \{(A,a)\ ,\ (A,bB)\ ,\ (B,a)\ ,\ (B,bA)\ ,\ (C,a)\}$ que l' on représente par :

$$G \left|\begin{array}{l} A \rightarrow a + bB \\ B \rightarrow a + bA \\ C \rightarrow a \end{array}\right. \qquad \text{où } \{a,b\} \subset \Sigma \text{ et } \mathbf{N} = \{A,B,C\}$$

on a $T_A(AAA,BBB) = \{(AA,BB)\ ,\ (BAA,ABB)\}$

$T_A(A,C) = \emptyset$ car $R(A,b) = bB$ et $R(C,b) = \perp$

Remarque : $T_A(\perp,\perp) = \{(\perp,\perp)\}$

On étend l' opération T_A sur $P(\mathbf{N}^*\times\mathbf{N}^*)$ par :

$$\forall\ E \subset \mathbf{N}^*\times\mathbf{N}^*,\ T_A(E) = \left|\begin{array}{l} \emptyset \text{ si } E = \emptyset \text{ ou } \exists\ (\alpha,\beta) \in E/\ T_A(\alpha,\beta) = \emptyset \\ \bigcup_{(\alpha,\beta)\in E} T_A(\alpha,\beta) \text{ sinon} \end{array}\right.$$

T_A (ou plus exactement son graphe) est une relation binaire sur $P(\mathbf{N}^*\times\mathbf{N}^*)$ et donc $(T_A)_n$ pour $n \in \mathbb{N}$, notée T_A^n , est parfaitement défini (cf. 2.2) .

Autrement dit , T_A^n est l' application composant T_A n fois .

Comme pour $\alpha,\ \beta \in \mathbf{N}^*$, $T_A(\{(\alpha,\beta)\}) = T_A(\alpha,\beta)$, on notera pour $n \in \mathbb{N}$

$T_A^n(\alpha,\beta)$ à la place de $T_A^n(\{(\alpha,\beta)\})$

b) la transformation de découpage gauche (le plus à gauche), notée T_B , opérant de $\mathbf{N}.\mathbf{N}^+ \times \mathbf{N}.\mathbf{N}^+$ dans $P(\mathbf{N}^*\times\mathbf{N}^*)$ mais qui n' est pas une application , est définie $\forall\ \alpha,\beta \in \mathbf{N}.\mathbf{N}^+$ par :

$$T_B(\alpha,\beta) = \begin{cases} \{(\alpha(1),\beta(1)\gamma)\,,\,(\gamma(\alpha\backslash 2),\beta\backslash 2)\} \text{ si } \tau_G(\alpha(1)) \geqslant \tau_G(\beta(1)) \\ \qquad\qquad \text{avec } (\gamma,\perp) \in T_A(\alpha(1),\beta(1)) \, \tau_G(\beta(1)) \\ (T_B(\beta,\alpha))^{-1} \text{ si } \tau_G(\alpha(1)) < \tau_G(\beta(1)) \\ \emptyset \text{ sinon} \end{cases}$$

Exemple : en prenant comme grammaire simple et réduite

$$G \quad \begin{cases} A \rightarrow a + bAB + c \\ B \rightarrow aA + bBB + cC \\ C \rightarrow a \end{cases}$$

$T_B(BA,AB) = (T_B(AB,BA))^{-1}$ car $\tau_G(A) = 1 < \tau_G(B) = 2$

$= \{(AA,B)\,,\,(B,AA)\}$ car $(\perp,A) \in T_A(A,B)$

On remarquera que $T_B(BA,AB) = \{(CA,B)\,,\,(B,AC)\}$ convient aussi

$T_B(A,C) = \emptyset$ car $T_A(A,C) = \emptyset$

On dit qu' une grammaire G est préfixe $\Leftrightarrow \forall A \in \mathbf{N}$, $L(G,A)$ est préfixe

Remarques G est préfixe $\Leftrightarrow \forall U \in (\Sigma \cup \mathbf{N})^*$, $L(G,U)$ est préfixe

toute grammaire simple est préfixe

On se dote de deux opérateurs

Div , dit de divergence, défini pour G grammaire quelconque par

$$\forall U,V \in (\Sigma\cup\mathbf{N})^*,\ \mathrm{Div}(U,V) = \begin{cases} \min\{\,|u| \,/\, u \in (L(G,U) - L(G,V)) \cup (L(G,V) - L(G,U))\,\} \text{ si } U \not\equiv_G V \\ \infty \text{ sinon} \end{cases}$$

et qui possède les propriétés suivantes

$\forall U,V \in (\Sigma\cup\mathbf{N})^*$, $U \not\equiv_G V \Leftrightarrow \mathrm{Div}(U,V) < \infty$

$\mathrm{Div}(U,V) = \mathrm{Div}(V,U)$

$\forall U,V,W \in (\Sigma\cup\mathbf{N})^*$, $\min(\mathrm{Div}(U,V),\mathrm{Div}(V,W)) \leqslant \mathrm{Div}(U,W)$

$\forall U,V,W,W' \in (\Sigma\cup\mathbf{N})^* \,/\, W \equiv_G W'$

$\mathrm{Div}(U,V) + \tau_G(W) \leqslant \mathrm{Div}(WU,W'V)$ et $\mathrm{Div}(U,V) + \tau_G(W) \leqslant \mathrm{Div}(UW,VW')$

Si G est préfixe alors les 2 inégalités sont des égalités

Divg , dit de divergence gauche, défini pour G grammaire préfixe par

$$\forall U,V \in (\Sigma\cup\mathbf{N})^*,\ \mathrm{Divg}(U,V) = \begin{cases} \min\{\,|u| \,/\, u \in (FG(L(G,U)) - FG(L(G,V))) \cup (FG(L(G,V)) - FG(L(G,U)))\,\} \text{ si } U \not\equiv_G V \\ \infty \text{ sinon} \end{cases}$$

c.a.d. à toute paire d' axiomes , engendrant une paire de langages préfixes distincts , Divg associe la plus petite longueur des mots facteur gauche d' un des langages sans l' être de l' autre .

Divg est bien défini i.e. $\forall$ U,V $\in$ $(\Sigma \cup N)^*$, U $\not\equiv_G$ V $\Rightarrow$ FG (L(G,U)) $\neq$ FG (L(G,V))
et de plus $\forall$ U,V $\in$ $(\Sigma \cup N)^*$, Divg(U,V) $\leqslant$ Div(U,V)

Divg possède les mêmes propriétés (énoncées ci-dessus) que Div excepté que

$$\forall\ U,V,W,W' \in (\Sigma \cup N)^* \ /\ W \equiv_G W'$$
$$1 \leqslant \mathrm{Divg}(U,V) \leqslant \mathrm{Divg}(UW,VW') \leqslant \mathrm{Divg}(U,V) + \tau_G(W) = \mathrm{Divg}(WU,W'V)$$

Les propriétés de Div et Divg sont établies en 5.1

Pour T , étant la T_A ou la T_B transformation , on note
Ker(T) le domaine de définition de T
c.a.d. $\mathrm{Ker}(T_A) = N^* \times N^*$ et $\mathrm{Ker}(T_B) = N.N^+ \times N.N^+$
on dit que

T est valide $\Leftrightarrow$ $\forall\ (\alpha,\beta) \in \mathrm{Ker}(T)$, ($\alpha \equiv_G \beta \Leftrightarrow T(\alpha,\beta) \neq \emptyset$ et $T(\alpha,\beta) \subset \equiv_G$)
T est monotone $\Leftrightarrow$ $\forall\ (\alpha,\beta) \in \mathrm{Ker}(T)$, ($\alpha \not\equiv_G \beta$ et $T(\alpha,\beta) \neq \emptyset \Rightarrow$
$\exists\ (\alpha',\beta') \in T(\alpha,\beta)\ /\ \mathrm{Div}(\alpha',\beta') < \mathrm{Div}(\alpha,\beta)$
)

et on montre en 5.2 que T_A et T_B sont des transformations valides et monotones.

2.5 Réduction

On introduit ici une nouvelle opération , de réduction sur $\mathbf{N}^*\times\mathbf{N}^*$ d' un élément relativement à une famille , qui nous permettra d' exprimer efficacement et aisément , pour toute grammaire simple et réduite , l' arrêt par conséquence théorique (cf. 1) .

$$\text{Soit } \prod(\mathbf{N}) = \left\{ \mathrm{A} \subset \mathbf{N}\times\mathbf{N}^+ \;/\; \begin{array}{l} (1)\ \forall (\alpha,\beta) \in \mathrm{A}\,,\ \tau_G(\alpha) = \tau_G(\beta) \\ (2)\ \forall (\alpha,\beta), (\alpha',\beta') \in \mathrm{A}\,,\ \left| \begin{array}{l} \alpha \neq \alpha' \text{ si } \beta \neq \beta' \\ \text{et } \beta' \neq \alpha \end{array} \right. \end{array} \right\}$$

alors $\forall\, \mathrm{A} \in \prod(\mathbf{N})$, A est un système de réécriture canonique , c' est à dire noethérien et confluent (la preuve est donnée en 5.3) .

A est noethérien signifie que $\underset{A}{\rightarrow}$ (cf. 2.2) n' admet pas de chaine infinie ou autrement dit que l' on ne peut pas dériver indéfiniment à partir d' un élément de $(\Sigma\cup\mathbf{N})^*$ à l' aide de $\underset{A}{\rightarrow}$

i.e. $\forall\, U \in (\Sigma\cup\mathbf{N})^*, \exists\, n_U \geqslant 0 \;/\; \mathrm{Im}(\{U\}, \overset{n_U}{\underset{A}{\rightarrow}}) = \emptyset$

ce qui implique que

$\forall\, U \in (\Sigma\cup\mathbf{N})^*, \exists\, U{\downarrow}\mathrm{A} \in (\Sigma\cup\mathbf{N})^* \;/\; U \overset{*}{\underset{A}{\rightarrow}} U{\downarrow}\mathrm{A}$ et $\mathrm{Im}(\{U{\downarrow}\mathrm{A}\}, \underset{A}{\rightarrow}) = \emptyset$

$U{\downarrow}\mathrm{A}$ est appelé une A-forme normale de U

A est confluent veut dire que l' on peut trouver un mot commun dérivant par A de deux autres mots si ces derniers dérivent par A d' un même mot

i.e. $\forall\, U\ (\Sigma\cup\mathbf{N})^*, \forall\, U', V' \in \mathrm{Im}(\{U\}, \overset{*}{\underset{A}{\rightarrow}})$,

$\exists\, V \in \mathrm{Im}(\{U'\}, \overset{*}{\underset{A}{\rightarrow}}) \cap \mathrm{Im}(\{V'\}, \overset{*}{\underset{A}{\rightarrow}})$

que l' on représente généralement à l' aide du diagramme suivant :

$$\begin{tikzcd} & U \arrow[dl, "*"', "A"] \arrow[dr, "*", "A"'] & \\ U' \arrow[dr, "A"', "*"] & & V' \arrow[dl, "*"', "A"] \\ & U & \end{tikzcd}$$

Par conséquent , pour A système canonique , toute A-forme normale $U{\downarrow}\mathrm{A}$ de tout terme U existe et est unique . (De plus , $\overset{*}{\underset{A}{\longleftrightarrow}}$ est décidable) .

On peut alors définir une application , notée S , de réduction relativement à une "théorie" , application de $(\mathbf{N}^*\times\mathbf{N}^*) \times \prod(\mathbf{N})$ dans $\mathbf{N}^*\times\mathbf{N}^*$, par :

$$\forall\, \Phi \in \mathbf{N}^*\times\mathbf{N}^*, \forall\, \mathrm{A} \in \prod(\mathbf{N})\,,\ S(\Phi,\mathrm{A}) = (\alpha{\downarrow}\mathrm{A}, \beta{\downarrow}\mathrm{A}) \quad \text{où} \quad \Phi = (\alpha,\beta)$$

Exemple : l' opération de réduction appliquée à (A,CE) relativement à

A = {(A,B), (B,CD), (E,D)} $\in \prod(\mathbf{N})$ donne S((A,CE), A) = (CD,CD)

3 Algorithme

3.1 Présentation de l' algorithme

Soit $(\alpha,\beta) \in \mathbf{N}^*\times\mathbf{N}^*$. L' algorithme de décision de $(\alpha,\beta) \in \equiv_G$ est celui de la construction d' un arbre sur $\mathbf{N}^*\times\mathbf{N}^*$ non gradué , dit arbre d' équivalence et noté $\mathrm{Arbre}_G\ (\alpha,\beta) \in \mathrm{P}(\mathbf{N}_+^*\times(\mathbf{N}^*\times\mathbf{N}^*))$ (graphe de l' arbre) , que l' on définit inductivement par ordre lexicographique comme suit :

> $\mathrm{Arbre}_G\ (\alpha,\beta) = \mathrm{Const}\ (\{(\perp,(\alpha,\beta))\}, \perp, \emptyset)$
> où Const est une application de $\mathrm{P}(\mathbf{N}_+^*\times(\mathbf{N}^*\times\mathbf{N}^*)) \times \mathbf{N}_+^* \times \prod(\mathbf{N})$ dans $\mathrm{P}(\mathbf{N}_+^*\times(\mathbf{N}^*\times\mathbf{N}^*))$ qui à T, u, A [désignant respectivement l' arbre , le noeud de l' arbre en cours , le système de réécriture en cours] , associe Const(T, u, A) défini dans l' ordre par les 5 règles suivantes :

remarque : on utilise le caractère $\leftarrow$ comme symbole d' affectation

a) Si les deux éléments du label en cours sont de valuations différentes alors on s' arrête i.e.

$(\lambda,\mu) \leftarrow \mathrm{T}(u)$

Si $\tau_G(\lambda) \neq \tau_G(\mu)$ **Alors** T **fs**

b) On réduit le label en cours à l' aide du système courant et on le rajoute dans l' arbre (plus lisible) si c' est un nouvel élément c.a.d.

$(\lambda',\mu') \leftarrow \mathrm{S}(\mathrm{T}(u),\mathrm{A})$

Si $(\lambda',\mu') \neq (\lambda,\mu)$ **Alors** $\mathrm{Dom}(\mathrm{T}) \leftarrow \mathrm{Dom}(\mathrm{T}) \cup \{u1\}$; $\mathrm{T}(u1) \leftarrow (\lambda',\mu')$; $u \leftarrow u1$ **fs**

c) Si le label en cours est réflexif alors son noeud est terminal et on passe au noeud suivant (ordre lexicographique) s' il y en a un c.a.d.

Si $\lambda' = \mu'$ **Alors**

$\mathrm{B} \leftarrow \{v \in \mathrm{Dom}(\mathrm{T}) / u <_{lex} v\}$

Si $\mathrm{B} = \emptyset$ ***Alors*** T ***Sinon*** Const $(\mathrm{T}, \min_{\leq_{lex}} \mathrm{B}, \mathrm{A})$ ***fs***

fs

d) Si les longueurs des deux mots (de non-terminaux) du label en cours sont au moins égales à 2 alors on essaye de découper ces mots à l' aide d' une $\mathrm{T_B}$ transformation . Si le découpage échoue alors on s' arrête sinon on continue avec la "paire découpée la plus à gauche" c.a.d.

Si $\min(|\lambda'|,|\mu'|) > 1$ **Alors**

$\mathrm{B} \leftarrow \mathrm{T_B}(\lambda',\mu')$

Si $\mathrm{B} = \emptyset$ ***Alors*** T ***Sinon***

Dom$(\mathrm{T}) \leftarrow \mathrm{Dom}(\mathrm{T}) \cup \{u1,u2\}$

$\{\mathrm{T}(u1),\mathrm{T}(u2)\} \leftarrow \mathrm{B}$ en choisissant pour $\mathrm{T}(u1)$ la paire $(\lambda'(1),\mu'(1)\gamma)$ si

$\mathfrak{f}$

$\tau_G(\lambda'(1)) \geqslant \tau_G(\mu'(1))$ ou $(\lambda'(1)\gamma, \mu'(1))$ sinon (avec $\gamma \in \mathbf{N}^*$)

Const (T, u1, A)

fs

fs

e) Si un des mots du label est de longueur au plus égale à 1 , alors on essaye de faire un développement en parallèle à l' aide d' une T_A transformation . Si l' opération échoue alors on s' arrête sinon on rajoute le label dans le système (de réécriture) courant et on passe au noeud développé le plus à gauche c.a.d.

Si $\min(|\lambda'|, |\mu'|) \leqslant 1$ **Alors**

$B \leftarrow T_A(\lambda', \mu')$

Si $B = \emptyset$ ***Alors*** T ***Sinon***

Dom(T) ← Dom(T) U $\{u\}.[\mathrm{Card}(B)]$

$\{T(u1), \ldots, T(u.\mathrm{Card}(B))\} \leftarrow B$ de disposition libre

Si $|\lambda'| = 1$ Alors $A \leftarrow A \cup \{(\lambda', \mu')\}$ Sinon $A \leftarrow A \cup \{(\mu', \lambda')\}$ fs

Const (T, u1, A)

fs

fs

On montre (cf. 5.4) que

l' algorithme est bien défini et s' arrête toujours (Const ne "boucle" pas) i.e. que l' algorithme génère , pour tout couple de mots non terminaux , un arbre d' équivalence fini .

et que

$\forall \alpha, \beta \in \mathbf{N}^*, \ \alpha \equiv_G \beta \Leftrightarrow$ les feuilles de $\mathrm{Arbre}_G(\alpha, \beta)$ sont toutes des paires réflexives

On remarquera que l' algorithme peut être traduit uniquement en terme de listes sans utiliser d' arbre mais on perd la représentation graphique (arborescente) de l' algorithme .

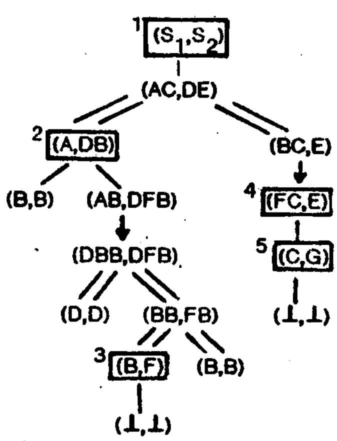
3.2 Exemples

On représente les opérations T_A , T_B , S respectivement par une barre, une double barre et une flèche. Les éléments (à une symétrie près) du système de réduction (réécriture) sont encadrés et numérotés par ordre croissant d' apparition (c.a.d. par ordre lexicographique).

Exemple 1 : on reprend un exemple de Korenjak-Hopcroft

$$G \quad \left| \begin{array}{l} S_1 \rightarrow aAC \\ A \rightarrow aB + bAB \\ B \rightarrow b \\ C \rightarrow a \end{array} \right. \qquad U \qquad \left| \begin{array}{l} S_2 \rightarrow aDE \\ D \rightarrow a + bDF \\ E \rightarrow bG \\ F \rightarrow b \\ G \rightarrow a \end{array} \right.$$

L' algorithme appliqué à (S_1,S_2) génère l' arbre, $Arbre_G$ (S_1,S_2) , suivant :

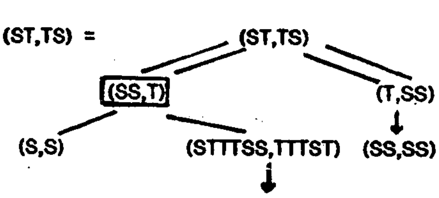


Toutes les feuilles de l' arbre étant réflexives, on a donc $S_1 \equiv_G S_2$

Exemple 2 : soit la grammaire simple et réduite suivante

$$G \quad \left| \begin{array}{l} S \rightarrow a + bSTTTS \\ T \rightarrow aS + bTTTST \end{array} \right. \qquad \text{donc} \quad \begin{array}{l} \mathcal{T}_G(S) = 1 \\ \mathcal{T}_G(T) = 2 \end{array}$$

$Arbre_G$ (ST,TS) =



d' où $ST \equiv_G TS$

Donnons encore deux exemples mais pour lesquels il n' y a pas équivalence

Exemple 3 :

$$\text{soit } G \quad \left| \begin{array}{l} S_1 \to a \\ S_2 \to b \\ S_3 \to aS_4S_5 + bS_1 \\ S_4 \to aS_3S_5 + bS_2 \\ S_5 \text{ quelconque (mais tel que } G \text{ reste simple et réduite)} \end{array} \right.$$

$\text{Arbre}_G\,(S_3,S_4) =$

- $\boxed{(S_3,S_4)}$
  - (S_4S_5,S_3S_5)
    - $\downarrow\ (S_4S_5,S_4S_5)$
  - (S_1,S_2)

La feuille non-réflexive (S_1,S_2) indique que $S_3 \not\equiv_G S_4$. On remarquera qu' en conséquence $S_4S_5 \not\equiv_G S_3S_5$ mais pourtant cette paire a été réduite en une paire équivalente (réflexive).

Exemple 4 :

$$G \quad \left| \begin{array}{l} S_1 \to a \\ S_2 \to b \\ S_3 \to a + bS_4S_5 + cS_7 \\ S_4 \to a + bS_3S_6 + cS_8 \\ S_5 \to a + bS_6S_5S_4S_2 \\ S_6 \to a + bS_5S_6S_3S_1 \\ S_7 \text{ et } S_8 \quad \text{quelconques} \end{array} \right.$$

- $\boxed{(S_3,S_4)}$
  - $(\perp,\perp)$
  - (S_4S_5,S_3S_6)
    - $\downarrow\ (S_4S_5,S_4S_6)$
      - (S_4,S_4)
      - $\boxed{(S_5,S_6)}$
        - $(\perp,\perp)$
        - $(S_6S_5S_4S_2,S_5S_6S_3S_1)$
          - $\downarrow\ (S_6S_6S_4S_2,S_6S_6S_4S_1)$
            - (S_6,S_6)
            - $(S_6S_4S_2,S_6S_4S_1)$
              - (S_6,S_6)
              - (S_4S_2,S_4S_1)
                - (S_4,S_4)
                - (S_2,S_1)
  - (S_7,S_8)

Ainsi, $S_3 \not\equiv_G S_4$. On remarquera que l' on peut ajouter, au développement de l' arbre, une opération de simplification mais ceci ne diminue en rien la complexité de l' algorithme.

4 Critère d' équivalence

4.1 Présentation du critère

La complexité de l' algorithme , liée à la taille de l' arbre d' équivalence , est déterminée (comme déjà indiqué par Korenjak-Hopcroft) en comparant pour tout couple d' axiomes (i.e. pour tout élément de $\mathbf{N}^*\times\mathbf{N}^*$) l' arbre fini d' équivalence avec l' arbre de transformation parallèle gauche qui est l' arbre , généralement infini , obtenu en développant uniquement par T_A transformation . Cette comparaison permet la connaissance d' une limite minimale sur la hauteur de l' arbre de dérivation parallèle gauche , dépendant linéairement de caractéristiques de la grammaire et du couple d' axiomes , après laquelle il n' est plus nécessaire de développer pour décider de l' équivalence (des axiomes) .

Soient les caractéristiques , de la grammaire G , suivantes :

$$|G| = \mathrm{Card}\ (\mathbf{N}) \quad \text{la taille de } G$$
$$\forall A \in \mathbf{N},\ e(A) = \max_{\mathcal{T}_G} (\mathrm{Im}(\{A\},G)) - 1$$
$$e = \max_{A\in\mathbf{N}} (e(A)) \quad \text{i.e.} \quad e = \max_{\mathcal{T}_G} (\mathrm{Im}(\mathbf{N},G)) - 1$$

Le critère d' équivalence est le suivant :

Soient $\alpha, \beta \in \mathbf{N}^*$. Notons
$$n_{\alpha,\beta} = \begin{cases} \min(\mathcal{T}_G(\alpha), \mathcal{T}_G(\beta)) + 1 & \text{si } \mathcal{T}_G(\alpha) \neq \mathcal{T}_G(\beta) \\ (|G|-1)e + 1 & \text{si } \mathcal{T}_G(\alpha) = \mathcal{T}_G(\beta),\ \min(|\alpha|,|\beta|) = 1,\ |G| > 2 \text{ ou } e = 0 \\ e + 2 & \text{si } \mathcal{T}_G(\alpha) = \mathcal{T}_G(\beta),\ \min(|\alpha|,|\beta|) = 1,\ |G| = 2 \text{ et } e \neq 0 \\ (|G|-1)e + \mathcal{T}_G(\alpha) + 1 & \text{si } \mathcal{T}_G(\alpha) = \mathcal{T}_G(\beta),\ \min(|\alpha|,|\beta|) > 1,\ |G| > 2 \text{ ou } e = 0 \\ e + \mathcal{T}_G(\alpha) + 2 & \text{si } \mathcal{T}_G(\alpha) = \mathcal{T}_G(\beta),\ \min(|\alpha|,|\beta|) > 1,\ |G| = 2 \text{ et } e \neq 0 \end{cases}$$

$$\alpha \equiv_G \beta \Leftrightarrow (FG(L(G,\alpha)) \cap \Sigma^{n_{\alpha,\beta}} = FG(L(G,\beta)) \cap \Sigma^{n_{\alpha,\beta}})$$

Remarque 1 : $\forall\ \alpha, \beta \in \mathbf{N}^*,\ n_{\alpha,\beta} \leqslant \min(\mathcal{T}_G(\alpha), \mathcal{T}_G(\beta)) + (|G|-1)e + 2$

Remarque 2 : Si on reprend les caractéristiques habituelles pour G , à savoir :

$$r = \max_{|\,|} (\mathrm{Im}(\mathbf{N},G)) - 1$$
$$\ell = \max_{\mathcal{T}_G} (\mathbf{N}) \geqslant 1 \quad \text{car } G \text{ est } \bot\text{-libre}$$

on a

$$\ell = \max_{A\in\mathbf{N}} (\mathcal{T}_G(A)) = \max_{A\in\mathbf{N}} \min_{\mathcal{T}_G} \mathrm{Im}(\{A\},G) \leqslant \max_{A\in\mathbf{N}} (e(A)+1) = e + 1$$

de plus , par définition de ℓ , $\forall A \in \mathbf{N}\ \ \mathcal{T}_G(A) \leqslant \ell$

et comme $\mathcal{T}_G(\alpha\beta) = \mathcal{T}_G(\alpha) + \mathcal{T}_G(\beta)$ pour $\alpha,\beta \in \mathbf{N}^*$

donc , par induction sur $|\alpha|$ pour $\alpha \in \mathbf{N}^*$, $\mathcal{T}_G(\alpha) \leqslant \ell\,|\alpha|$

d' où , G étant en FNG , $\forall A \in \mathbf{N}$, $e(A) \leqslant \ell(\max_{||}(Im(\{A\},G) - 1)$

par conséquent , $e \leqslant \ell(\max_{||}(Im(\mathbf{N},G) - 1) = \ell r$

ainsi , $0 \leqslant \ell - 1 \leqslant e \leqslant \ell r$

il en résulte que pour $r = 2$, $|\alpha| = |\beta| = 1$ et $\mathcal{T}_G(\alpha) = \mathcal{T}_G(\beta)$

$n_{\alpha,\beta} \leqslant 2\ell(|G| - 1) + 1$ (+ 2 au lieu de + 1 si $|G|=2$ et $e \neq 0$)

qu' il faut comparer avec

$\ell |G|^{(\ell+1)(\ell+3)}$ de Korenjak-Hopcroft

$\ell |G|^{2(\ell+1)}$ de Wood

$2\ell |G|^2$ de Butzbach

On montre en 4.2 que , pour toute grammaire simple et réduite , $n_{\alpha,\beta}$ est optimal $\forall \alpha, \beta \in \mathbf{N}^*$

Remarque 3 : Ce critère devient préférable pour décider de l' équivalence des langages simples . De plus , il est adapté au traitement vectoriel puisqu' il revient à générer l' arbre de transformation parallèle gauche de hauteur minimale pour la décision . Cependant , c' est l' algorithme qui , d' une part permet de valider le critère et , d' autre part est propre à une généralisation .

4.2 Optimalité du critère

On établit l' optimalité du critère en montrant que pour toutes valeurs p, q, r, s des paramètres d' expression de $n_{\alpha,\beta}$ ($|G|$, e, $\mathcal{T}_G(\alpha)$, $\mathcal{T}_G(\beta)$) , on peut trouver une grammaire G simple et réduite et $\alpha, \beta \in (\mathrm{Ker}(G))^*$ /

$$|G| = p\ ,\ e = q\ ,\ \mathcal{T}_G(\alpha) = r\ ,\ \mathcal{T}_G(\beta) = s \quad \text{et} \quad \mathrm{Divg}(\alpha,\beta) = n_{\alpha,\beta}$$

Bien entendu , l' optimalité dépend des paramètres choisis pour exprimer la limite .

L' optimalité de $n_{\alpha,\beta}$ pour $\mathcal{T}_G(\alpha) \neq \mathcal{T}_G(\beta)$ est établie par

Proposition 4.2.1 :

$\forall r \geqslant 0$, $\forall$ G grammaire simple et réduite , $\exists\ \alpha, \beta \in (\mathrm{Ker}(G))^*$ tels que

$$\mathcal{T}_G(\alpha) = r < \mathcal{T}_G(\beta) \quad \text{et} \quad \mathrm{Divg}(\alpha,\beta) = \min(\mathcal{T}_G(\alpha), \mathcal{T}_G(\beta)) + 1$$

Preuve : Comme G est réduite et $\perp$-libre , $\exists\ A \in \mathrm{Ker}(G)$ / $\mathcal{T}_G(A) = 1$

il suffit donc de prendre $\alpha = A^r$ et $\beta = A^{r+1}$ vu que

$\mathrm{Divg}(\alpha,\beta) = \mathcal{T}_G(A^r) + \mathrm{Divg}(\perp,A)$,d' après 5.1.4

$= r + 1$

$= \min(\mathcal{T}_G(\alpha), \mathcal{T}_G(\beta)) + 1$ □

L' optimalité de $n_{\alpha,\beta}$ pour $\mathcal{T}_G(\alpha) = \mathcal{T}_G(\beta)$, $\min(|\alpha|, |\beta|) = 1$ et $\alpha \not\equiv_G \beta$ est obtenue par

Proposition 4.2.2 :

$\forall (p,q) \in (\mathbb{N}_+ - \{1\}) \times \mathbb{N}$, $\exists$ G grammaire simple et réduite ,

$\exists\ S \in \mathrm{Ker}(G)$, $\exists\ T \in (\mathrm{Ker}(G))^*$ avec $\mathcal{T}_G(S) = \mathcal{T}_G(T)$ et $S \not\equiv_G T$ tels que

$|G| = p$, $e = q$ et $\mathrm{Divg}(S,T)$ = **Si** $p>2$ ou $q=0$ **Alors** $(p-1)q + 1$ **Sinon** $q + 2$ **fs**

Remarque : le cas p = 1 est exclu car

si G est une grammaire simple et réduite / $\mathrm{Card}(\mathrm{Ker}(G)) = 1$

alors $\forall\ \alpha,\beta \in (\mathrm{Ker}(G))^*$, $\alpha \not\equiv_G \beta \Leftrightarrow \mathcal{T}_G(\alpha) \neq \mathcal{T}_G(\beta)$

on est donc ramené au cas précédent

Preuve : i) Cas où q = 0 : soit G une grammaire simple et réduite / $|G| = p \geqslant 2$ et $q = 0$

c.a.d. tels que $\mathrm{Card}(\mathrm{Ker}(G)) = p$ et $\mathrm{Im}(\mathrm{Ker}(G),G) \subset \Sigma$

donc $\forall\ A \in \mathrm{Ker}(G)$, $\mathcal{T}_G(A) = 1$ alors

$\forall\ S,T \in \mathrm{Ker}(G)$, $S \not\equiv_G T \Leftrightarrow \mathrm{Divg}(S,T) = 1 = (p-1)q + 1$

ii) Cas où $p \geqslant 2$ et $q \geqslant 1$: soit $G_{p,q}$ la grammaire de représentation

$$G \quad \begin{cases} A_n \rightarrow a + b\lambda A_{n+1} & \text{pour } n \in [p-1] \\ A_p \rightarrow a + b\lambda & \end{cases} \quad \text{où } a,b \in \Sigma$$

avec $\lambda \in (\mathrm{Ker}(G))_*$ et $|\lambda| = q - 1 \geqslant 0$

ainsi , $\mathrm{Ker}(G) = \{A_1, \dots A_p\}$ donc $|G| = p$

de plus , $\forall\ n \in [p]$ $\mathcal{T}_G(A_n) = 1 \Rightarrow \mathcal{T}_G(\lambda) = |\lambda| = q - 1$

donc $\forall\, \eta \in [p-1]\quad e(A_\eta) = q$ et $e(A_p) = q-1 \Rightarrow e = q$

Posons $S = A_1$ et $T = A_2$. On a bien $\mathcal{T}_G(S) = \mathcal{T}_G(T)$

Montrons, par induction sur $p \geqslant 2$, que $S \not\equiv_G T$ et $Divg(S,T) = (p-1)q + 1$

Si $p = 2$, $Divg(S,T) = 1 + Divg(\lambda A_p, \lambda)$

$= 1 + \mathcal{T}_G(\lambda) + Divg(A_p, \perp)$

$= q + 1$

$= (p-1)q + 1$ donc $S \not\equiv_G T$

Supposons la propriété vraie pour $p \geqslant 2$ et montrons-là pour $p+1 = |G|$

$Divg(S,T) = Divg(A_1, A_2)$ par définition de S et T

$= 1 + Divg(\lambda A_2, \lambda A_3)$ par définition de A_1 et A_2

$= 1 + \mathcal{T}_G(\lambda) + Divg(A_2, A_3)$

$= q + Divg(A_2, A_3)$

Soit G' la grammaire obtenue à partir de G en supprimant le non-terminal A_1 c.a.d.

$$\left| \begin{array}{l} Ker(G') = Ker(G) - \{A_1\} \\ \forall A \in Ker(G'),\ Im(\{A\}, G') = Im(\{A\}, G) \end{array} \right.$$

G' est une grammaire $G_{p,q}$ et par hypothèse de récurrence, on a

$$Divg_{G'}(A_2, A_3) = (p-1)q + 1$$

d' où, par définition de G', $Divg_G(A_2, A_3) = (p-1)q + 1$

par conséquent, $Divg(S,T) = q + (p-1)q + 1 = pq + 1$ et $S \not\equiv_G T$

la récurrence est établie et ii) par la même occasion

iii) Cas où $p = 2$ et $q \geqslant 1$: soit G_q la grammaire de représentation

$$G \quad \left| \begin{array}{l} A_1 \rightarrow aA_2 + b\lambda A_2 \\ A_2 \rightarrow a + b\lambda A_2 \end{array} \right. \quad \text{où } a,b \in \Sigma$$

avec $\lambda \in (Ker(G))^* \ / \ \mathcal{T}_G(\lambda) = q - 1$

$\mathcal{T}_G(A_2) = 1$ donc $e = q$

Posons $S = A_1$ et $T = A_2A_2$. On a bien $\mathcal{T}_G(S) = \mathcal{T}_G(T)$

De plus, $Divg(S,T) = 1 + Divg(\lambda A_2, \lambda A_2 A_2)$

$= 1 + \mathcal{T}_G(\lambda) + \mathcal{T}_G(A_2) + Divg(\perp, A_2)$

$= q + 2$

iv) i) à iii) établissent la proposition 4.2.2 □

L' optimalité de $n_{\alpha,\beta}$ pour $\mathcal{T}_G(\alpha) = \mathcal{T}_G(\beta)$, $\min(|\alpha|, |\beta|) > 1$ et $\alpha \not\equiv_G \beta$ est immédiate à partir de G de ii) ou iii) en prenant comme paire d' axiomes

$$(A_p^i S A_p^{r-i}, A_p^i T A_p^{r-i}) \quad \text{où } r \geqslant 1 \text{ et } i \in \{0\} \cup [r]$$

5 Validité de l' algorithme

5.1 Propriétés de Div et Divg

On se propose de véfifier les propriétés énoncées en 2.4 des opérateurs Div et Divg .
Les propriétés suivantes de Div sont établies pour G quelconque.

Propriété 5.1.1 : $\forall$ U,V,W $\in (\Sigma \cup N)^*$, min (Div(U,V),Div(V,W)) $\leqslant$ Div(U,W)

Preuve : soit $U \equiv_G W$ alors Div(U,W) = ∞ d' où l' inégalité

soit $U \not\equiv_G W$ alors par définition de Div

soit $\exists$ u $\in$ L(G,U) - L(G,W) et | u | = Div(U,W)

si u $\in$ L(G,V) alors u $\in$ L(G,V) - L(G,W)

donc Div(V,W) $\leqslant$ | u | = Div(U,W)

si u $\notin$ L(G,V) alors u $\in$ L(G,U) - L(G,V)

donc Div(U,V) $\leqslant$ | u | = Div(U,W)

soit $\exists$ u $\in$ L(G,W) - L(G,U) et | u | = Div(U,W)

cas symétrique au précédent

la propriété 5.1.1 est donc établie □

Propriété 5.1.2 : $\forall$ U,V,W,W' $\in (\Sigma \cup N)^*$ / $W \equiv_G W'$

Div(U,V) + $\mathcal{T}_G$(W) $\leqslant$ Div(WU,W'V) et Div(U,V) + $\mathcal{T}_G$(W) $\leqslant$ Div(UW,VW')

Si G est préfixe alors Div(U,V) + $\mathcal{T}_G$(W) = Div(WU,W'V) = Div(UW,VW') .

Preuve :

i) Montrons que Div(U,V) + $\mathcal{T}_G$(W) $\leqslant$ Div(WU,W'V) ?

soit $WU \equiv_G W'V$ alors Div(WU,W'V) = ∞ et l' inégalité est vraie

soit $WU \not\equiv_G W'V$. Par symétrie, on limite l' étude à

$\exists$ u $\in$ L(G,WU) - L(G,W'V) / | u | = Div(WU,W'V)

comme u $\in$ L(G,WU) alors $\exists$ u' $\in$ L(G,W) , u'' $\in$ L(G,U) / u = u'u''

mais u' $\in$ L(G,W') et u = u'u'' $\notin$ L(G,W'V) donc u'' $\notin$ L(G,V)

par conséquent Div(U,V) $\leqslant$ | u'' | = | u | - | u' | $\leqslant$ Div(U,V) - $\mathcal{T}_G$(W)

d' où i)

ii) Montrons que Div(U,V) + $\mathcal{T}_G$(W) $\leqslant$ Div(UW,VW') ?

Se démontre de façon analogue à i)

iii) Montrons que $Div(WU,W'V) \leqslant Div(U,V) + \tau_G(W)$ si G est préfixe ?

soit $U \equiv_G V$ ou $L(G,W) = \emptyset$ alors $Div(U,V) + \tau_G(W) = \infty$ d' où iii)

soit $U \not\equiv_G V$ et $L(G,W) \neq \emptyset$. Par symétrie, on peut se restreindre à

$\exists$ u $\in$ L(G,U) - L(G,V) / | u | = Div(U,V)

Comme $L(G,W) \neq \emptyset$, on peut choisir un w $\in$ L(G,W) / | w | = $\tau_G(W)$

donc wu $\in$ L(G,WU)

or $W \equiv_G W'$ donc w $\in$ L(G,W') et | w | = $\tau_G(W')$

de plus L(G,W') est préfixe et u $\notin$ L(G,V)

par conséquent wu $\notin$ L(G,W'V)

ainsi $Div(WU,W'V) \leqslant |wu| = Div(U,V) + \tau_G(W)$

iii) est donc démontré

iv) On voit comme en iii) que $Div(UW,VW') \leqslant Div(U,V) + \tau_G(W)$ si G est préfixe . En définitive , la propriété 5.1.2 est entièrement démontrée □

On obtient alors le corollaire suivant

Pour toute grammaire G préfixe et réduite , $\equiv_G$ est simplifiable pour "."

Les propriétés suivantes sont établies pour G préfixe.

Propriété 5.1.3 : $\forall$ U,V $\in$ $(\Sigma \cup \mathbf{N})^*$, $U \not\equiv_G V \Leftrightarrow FG(L(G,U)) \neq FG(L(G,V))$

Preuve : i) Condition nécessaire : si $U \not\equiv_G V$ alors

soit $\exists$ u $\in$ L(G,U) / u $\notin$ L(G,V)

soit u $\notin$ FG (L(G,V))

or u $\in$ L(G,U) $\subset$ FG (L(G,U)) donc FG (L(G,U)) $\neq$ FG (L(G,V))

soit u $\in$ FG (L(G,V))

mais u $\notin$ L(G,V) | $\Rightarrow \exists$ a $\in \Sigma$ / ua $\in$ FG (L(G,V))

comme u $\in$ L(G,U) préfixe donc ua $\notin$ FG (L(G,U))

ainsi FG (L(G,U)) $\neq$ FG (L(G,V))

soit $\exists$ u $\in$ L(G,V) / u $\notin$ L(G,U)

cas symétrique au précédent

d' où i)

ii) La condition suffisante étant triviale , la propriété 5.1.3 est donc établie □

Propriété 5.1.4 : $\forall\ U,V,W,W' \in (\Sigma \cup N)^* \ / \ W \equiv_G W'$

$$1 \leqslant Divg(U,V) \leqslant Divg(UW,VW') \leqslant Divg(U,V) + \tau_G(W) = Divg(WU,W'V)$$

Preuve :

i) Montrons que $Divg(U,V) \leqslant Divg(UW,VW')$?

Si $UW \equiv_G VW'$ alors l' inégalité est triviale

Si $UW \not\equiv_G VW'$ alors

soit $\exists\ u \in FG(L(G,UW)) - FG(L(G,VW'))\ /\ |u| = Divg(UW,VW')$

si $u \in FG(L(G,U))$

comme $u \notin FG(L(G,VW')) \supset FG(L(G,V))$

alors $Divg(U,V) \leqslant |u| = Divg(UW,VW')$

si $u \notin FG(L(G,U))$

comme $u \in FG(L(G,UW))$

alors $\exists\ v \in L(G,U)\ ,\ w \in FG(L(G,W)) - \{\perp\}\ /\ u = vw$

soit $v \notin FG(L(G,V))$

or $v \in L(G,U) \subset FG(L(G,U))$

donc $Divg(U,V) \leqslant |v| < |u| = Divg(UW,VW')$

soit $v \in FG(L(G,V))$

donc $\exists\ w' \in \Sigma^*\ /\ vw' \in L(G,V)$

si $w' \neq \perp$

alors $vw'(1) \notin FG(L(G,U))$ car $v \in L(G,U)$ préfixe

de plus $vw'(1) \in FG(vw') \subset FG(L(G,V))$ donc

$Divg(U,V) \leqslant |vw'(1)| = |v|+1 \leqslant |u| = Divg(UW,VW')$

si $w' = \perp$ i.e. $v \in L(G,V)$

or $w \in FG(L(G,W)) = FG(L(G,W'))$ car $W \equiv_G W'$

donc $u = vw \in FG(L(G,VW'))$ exclu dans ce cas

soit $\exists\ u \in FG(L(G,VW')) - FG(L(G,UW))\ /\ |u| = Divg(UW,VW')$

cas symétrique au précédent

d' où i)

ii) Montrons que $Divg(UW,VW') \leqslant Divg(U,V) + \tau_G(W)$?

Si $U \equiv_G V$ ou $L(G,W) = \emptyset$ alors $Divg(U,V) + \tau_G(W) = \infty$

d' où l' inégalité

Si $\perp \in L(G,W)$ alors $L(G,W) = L(G,W') = \{\perp\}$ car G est préfixe

on a donc une égalité

Si $U \not\equiv_G V$ et $\perp \notin L(G,W) \neq \emptyset$ alors par raison de symétrie, on peut se contenter d' étudier le cas où

$\exists\ u \in FG(L(G,U)) - FG(L(G,V))\ /\ |u| = Divg(U,V)$

Soit $u \notin FG(L(G,VW'))$

comme $u \in FG(L(G,U)) \subset FG(L(G,UW))$

alors $Divg(UW,VW') \leqslant |u| = Divg(U,V) \leqslant Divg(U,V) + \mathcal{T}_G(W)$

Soit $u \in FG(L(G,VW'))$

or $u \notin FG(L(G,V))$ donc $u \neq \perp$ i.e. $|u| \geqslant 1$

posons alors u' le facteur gauche de u de longueur $|u|-1$

comme $u = u'.u(|u|) \in FG(L(G,U))$ avec $L(G,U)$ préfixe

donc $u' \in FG(L(G,U)) - L(G,U)$

de plus $u' \in FG(L(G,V))$ sinon $Divg(U,V) \leqslant |u'| = Divg(U,V) - 1$

mais $u = u'.u(|u|) \in FG(L(G,VW')) - FG(L(G,V))$

donc $\exists\ u'_1 \in FG(u') \ / \ u'_1 \in L(G,V)$

et comme $u' \in FG(L(G,V))$ avec $L(G,V)$ préfixe

alors $u' \in L(G,V)$

comme $L(G,W) \neq \emptyset$, prenons $w \in L(G,W) \ / \ |w| = \mathcal{T}_G(W)$

il s' en suit que $u'w \in L(G,VW')$. De plus

soit $u'w \notin FG(L(G,UW))$

donc $Divg(UW,VW') \leqslant |u'w| < |u| + |w| = Divg(U,V) + \mathcal{T}_G(W)$

soit $u'w \in FG(L(G,UW))$ alors $u'w \notin L(G,UW)$ d' après les 2 cas suivants

si $u'w \in FG(L(G,U))$ alors $u'w \notin L(G,UW)$ car $\perp \notin L(G,W)$

si $u'w \notin FG(L(G,U))$

alors $\exists\ w',w'' \in \Sigma^* \ / \ w = w'w''$ et $u'w' \in L(G,U)$

de plus $w' \neq \perp$ car $u' \notin L(G,U)$

donc $|w''| < |w| = \mathcal{T}_G(W) \Rightarrow w'' \notin L(G,W)$

alors $u'w = u'w'w'' \notin L(G,UW)$ préfixe

il en résulte que $Divg(UW,VW') \leqslant |u'w| + 1 = Divg(U,V) + \mathcal{T}_G(W)$

en définitive, ii) est établie

iii) Montrons que $Divg(U,V) + \mathcal{T}_G(W) \leqslant Divg(WU,W'V)$?

soit $WU \equiv_G W'V$ alors $Divg(WU,W'V) = \infty$ et l' inégalité est vraie

soit $WU \not\equiv_G W'V$. Par symétrie, on limite l' étude à

$\exists\ u \in FG(L(G,WU)) - FG(L(G,W'V)) \ / \ |u| = Divg(WU,W'V)$

comme $u \notin FG(L(G,W'V))$, on a $u \notin FG(L(G,W')) = FG(L(G,W))$

ainsi $u \in FG(L(G,WU)) - FG(L(G,W))$

d' où $\exists\ u' \in L(G,W)$, $u'' \in FG(L(G,U)) \ / \ u = u'u''$

mais $u' \in L(G,W')$ et $u = u'u'' \notin FG(L(G,W'V))$

donc $u'' \notin FG(L(G,V))$

par conséquent $Divg(U,V) \leqslant |u''| = |u| - |u'| \leqslant Divg(U,V) - \mathcal{T}_G(W)$

d' où iii)

iv) Montrons que $Divg(WU,W'V) \leqslant Divg(U,V) + \mathcal{T}_G(W)$?

soit $U \equiv_G V$ ou $L(G,W) = \emptyset$ alors $Divg(U,V) + \mathcal{T}_G(W) = \infty$ d' où iv)

soit $U \not\equiv_G V$ et $L(G,W) \neq \emptyset$. Par symétrie, on peut se restreindre à

$\exists\ u \in FG(L(G,U)) - FG(L(G,V))\ /\ |u| = Divg(U,V)$

Comme $L(G,W) \neq \emptyset$, on peut choisir un $w \in L(G,W)\ /\ |w| = \mathcal{T}_G(W)$

donc $wu \in FG(L(G,WU))$

De plus , $W \equiv_G W'$ donc $w \in L(G,W')$ préfixe

or $u \notin FG(L(G,V))$ il s' en suit que $wu \notin FG(L(G,W'V))$

Ainsi $Divg(WU,W'V) \leqslant |wu| = Divg(U,V) + \mathcal{T}_G(W)$

iv) est donc démontré

v) i) à iv) prouvent la proposition 5.1.4 □

Propriété 5.1.5 : $\forall\ U,V,W \in (\Sigma \cup \mathbf{N})^*$, $\min(Divg(U,V),Divg(V,W)) \leqslant Divg(U,W)$

la démonstration est similaire à celle de la proposition 5.1.1

Remarque :

Notons $E = (\Sigma \cup \mathbf{N})^* /_{\equiv_G}$ l' ensemble des classes d' équivalence sur $(\Sigma \cup \mathbf{N})^*$ pour l' équivalence $\equiv_G$

et posons $\forall\ C, C' \in E$, $\left| \begin{array}{l} d(C,C') = \dfrac{1}{Div(U,U')} \\ dg(C,C') = \dfrac{1}{Divg(U,U')} \end{array} \right.$ où $U \in C$, $U' \in C'$

alors d et dg sont des distances ultra-métriques sur E , dites de divergence et de divergence gauche .

5.2 Validité et monotonicité des transformations T_A et T_B

Les définitions ont été données en 2.4

Proposition 5.2.1 : Les transformations T_A et T_B sont monotones

Preuve : i) T_A est monotone ?

Prenons $(\alpha,\beta) \in \mathrm{Ker}(T_A) = \mathbf{N}^* \times \mathbf{N}^* \;/\; \alpha \not\equiv_G \beta$ et $T_A(\alpha,\beta) \neq \emptyset$

et montrons que $\exists\ (\alpha',\beta') \in T_A(\alpha,\beta) \;/\; \mathrm{Div}(\alpha',\beta') < \mathrm{Div}(\alpha,\beta)$

or $\alpha,\beta \in \mathbf{N}^+$ car

si $\alpha = \perp$ alors $\beta \neq \perp$ vu que $\alpha \not\equiv_G \beta$

et donc $T_A(\alpha,\beta) = \emptyset$ ce qui est exclu

on voit de même pour $\beta = \perp$

$\alpha \not\equiv_G \beta$ donc par raisons de symétrie, on peut se contenter d' étudier le cas où

$\exists\ u \in L(G,\alpha) - L(G,\beta) \;/\; |u| = \mathrm{Div}(\alpha,\beta)$ (1)

comme G est simple (donc $\perp$ libre) , $u \in L(G,\alpha)$, $\alpha \neq \perp$

alors $u \neq \perp$ et $\exists !\ \gamma \in \mathbf{N}^* \;/\; (\alpha(1),u(1)\gamma) \in G$ (2)

donc $R(\alpha,u(1)) = u(1)\gamma$ (R a été définie en2.4)

d' où $R(\beta,u(1)) \neq \perp$ car $T_A(\alpha,\beta) \neq \emptyset$

i.e. $\exists !\ \delta \in \mathbf{N}^* \;/\; (\beta(1),u(1)\delta) \in G$ (3)

(1), (2), (3) donnent $u\backslash 2 \in L(G,\gamma(\alpha\backslash 2)) - L(G,\delta(\beta\backslash 2))$

i.e. $\mathrm{Div}(\alpha',\beta') \leqslant |u\backslash 2| < |u| = \mathrm{Div}(\alpha,\beta)$ avec $(\alpha',\beta') = (\gamma(\alpha\backslash 2)\,,\,\delta(\beta\backslash 2))$

de plus $(\alpha',\beta') \in T_A(\alpha,\beta)$ d' après (2) et (3)

d' où i)

ii) T_B est monotone ?

Soit $(\alpha,\beta) \in \mathrm{Ker}(T_B) = \mathbf{N}\,.\,\mathbf{N}^+ \times \mathbf{N}\,.\,\mathbf{N}^+ \;/\; \alpha \not\equiv_G \beta$ et $T_B(\alpha,\beta) \neq \emptyset$

par symétrie, on peut supposer que $\tau_G(\alpha(1)) \geqslant \tau_G(\beta(1))$

d' où $\exists\ \gamma \in \mathbf{N}^* \;/\; T_B(\alpha,\beta) = \{\, (\alpha(1),\beta(1)\gamma)\,,\,(\gamma(\alpha\backslash 2),\beta\backslash 2)\,\}$

d' après la propriété 5.1.2 , on a

$$\mathrm{Div}(\alpha\,,\,\beta(1)\gamma(\alpha\backslash 2)) = \mathrm{Div}(\alpha(1)\,,\,\beta(1)\gamma) + \tau_G(\alpha\backslash 2)$$

$$\mathrm{Div}(\beta(1)\gamma(\alpha\backslash 2)\,,\,\beta) = \mathrm{Div}(\gamma(\alpha\backslash 2)\,,\,\beta\backslash 2) + \tau_G(\beta(1))$$

de plus , d' après la propriété 5.1.1

$$\min\ (\mathrm{Div}(\alpha\,,\,\beta(1)\gamma(\alpha\backslash 2))\,,\,\mathrm{Div}(\beta(1)\gamma(\alpha\backslash 2)\,,\,\beta)) \leqslant \mathrm{Div}(\alpha,\beta)$$

il en résulte que

$$\min\ (\mathrm{Div}(\alpha(1),\beta(1)\gamma)+\tau_G(\alpha\backslash 2)\,,\,\mathrm{Div}(\gamma(\alpha\backslash 2),\beta\backslash 2)+\tau_G(\beta(1))) \leqslant \mathrm{Div}(\alpha,\beta) \quad (1)$$

or $|\alpha\backslash 2|$, $|\beta(1)| \geqslant 1$ donc $\mathcal{T}_G(\alpha\backslash 2)$, $\mathcal{T}_G(\beta(1)) > 0$ car G est $\perp$ libre (2)

soit $\mathrm{Div}(\alpha(1),\beta(1)\gamma) = \infty$ i.e. $\alpha(1) \equiv_G \beta(1)\gamma$

alors (1) $\Rightarrow$ $\mathrm{Div}(\gamma(\alpha\backslash 2)\ ,\ \beta\backslash 2) + \mathcal{T}_G(\beta(1)) \leqslant \mathrm{Div}(\alpha,\beta)$ (3)

de plus $\alpha \equiv_G \beta(1)\gamma(\alpha\backslash 2)$ et $\alpha \not\equiv_G \beta = \beta(1)(\beta\backslash 2)$

donc $\gamma(\alpha\backslash 2) \not\equiv_G \beta\backslash 2$ car G est préfixe et réduite

i.e. $\mathrm{Div}(\gamma(\alpha\backslash 2)\ ,\ \beta\backslash 2) < \infty$ (4)

(2) , (3) , (4) donnent $\mathrm{Div}(\gamma(\alpha\backslash 2)\ ,\ \beta\backslash 2) < \mathrm{Div}(\alpha,\beta)$

soit $\mathrm{Div}(\gamma(\alpha\backslash 2)\ ,\ \beta\backslash 2) = \infty$

cas similaire au précédent

soit $\mathrm{Div}(\alpha(1),\beta(1)\gamma) < \infty$ et $\mathrm{Div}(\gamma(\alpha\backslash 2)\ ,\ \beta\backslash 2) < \infty$

(1) et (2) $\Rightarrow$ $\min\ (\mathrm{Div}(\alpha(1),\beta(1)\gamma)\ ,\ \mathrm{Div}(\gamma(\alpha\backslash 2),\beta\backslash 2)) \leqslant \mathrm{Div}(\alpha,\beta)$

dans tous les cas, ii) est satistait

iii) i) et ii) valident la proposition 5.2.1 □

Proposition 5.2.2 : Les transformations T_A et T_B sont valides

Preuve : i) D' après 5.2.1 , les transformations T_A et T_B étant monotones , il ne nous reste donc plus qu' á vérifier la condition nécessaire de la validité des transformations, á savoir

si T correspond á T_A ou T_B , $\forall\ (\alpha,\beta) \in \mathrm{Ker}(T) \cap \equiv_G$, $\emptyset \neq T(\alpha,\beta) \subset \equiv_G$

ii) Condition nécessaire de la validité de T_A ?

Prenons $(\alpha,\beta) \in \mathrm{Ker}(T_A) = \mathbf{N}^* \times \mathbf{N}^*\ /\ \alpha \equiv_G \beta$

soit $\alpha = \beta = \perp$ et donc $T_A(\alpha,\beta) = \{(\perp,\perp)\}$ d' où ii)

soit $\alpha,\beta \in \mathbf{N}^+$

comme G est en FNG , on a donc

$$L(G,\alpha) = \bigcup_{(\alpha(1),a\gamma_a)\in G} \{a\} \cdot L(G,\gamma_a(\alpha\backslash 2)) = L(G,\beta) = \bigcup_{(\beta(1),b\delta_b)\in G} \{b\} \cdot L(G,\delta_b(\beta\backslash 2))$$

avec $a,b \in \Sigma$, $\gamma_a,\delta_a \in \mathbf{N}^*$

or G est simple donc si $(\alpha(1),a\gamma_a) \in G$ alors γ_a est unique

de même , si $(\beta(1),b\delta_b) \in G$ alors δ_b est unique

d' où $\{a \in \Sigma\ /\ (\alpha(1),a\gamma_a) \in G\} = \{b \in \Sigma\ /\ (\beta(1),b\delta_b) \in G\}$ noté E

avec $\forall\ a \in E$, $L(G,\gamma_a(\alpha\backslash 2)) = L(G,\delta_a(\beta\backslash 2))$

et $T_A(\alpha,\beta) = \{(\gamma_a(\alpha\backslash 2),\delta_a(\beta\backslash 2))\ /\ a \in E\}$

ainsi ii) est démontré

iii) Condition nécessaire de la validité de T_B ?

soit $(\alpha,\beta) \in Ker(T_B) = \mathbf{N}.\mathbf{N}^+ \times \mathbf{N}.\mathbf{N}^+ \;/\; \alpha \equiv_G \beta$

on doit donc montrer , pour n' importe quel $T_B(\alpha,\beta)$ possible , que

$$\emptyset \neq T_B(\alpha,\beta) \subset \equiv_G$$

par symétrie , on peut restreindre l' étude au cas où $\mathcal{T}_G(\alpha(1)) \geqslant \mathcal{T}_G(\beta(1))$

or $\mathcal{T}_G(\beta(1)) < \infty$ car G est réduite

on peut donc réitérer ii) $\mathcal{T}_G(\beta(1))$ fois pour obtenir

$$\emptyset \not\subset T_A^{\mathcal{T}_G(\beta(1))}(\alpha,\beta) \subset \equiv_G \quad (1)$$

de plus , par définition de T_A et $\mathcal{T}_G$

$$T_A^{\mathcal{T}_G(\beta(1))}(\alpha,\beta) = \{ (\alpha'(\alpha\backslash 2),\beta'(\beta\backslash 2)) \;/\; (\alpha',\beta') \in T_A^{\mathcal{T}_G(\beta(1))}(\alpha(1),\beta(1)) \} \quad (2)$$

(1) et (2) $\Rightarrow$ $\emptyset \not\subset T_A^{\mathcal{T}_G(\beta(1))}(\alpha(1),\beta(1))$

et donc $T_A^{\mathcal{T}_G(\beta(1))}(\alpha(1),\beta(1)) \cap \mathbf{N}^* \times \{\bot\} \neq \emptyset$

d' où $T_B(\alpha,\beta) = \{ (\alpha(1),\beta(1)\gamma) , (\gamma(\alpha\backslash 2),\beta\backslash 2) \}$ avec $(\gamma,\bot) \in T_A^{\mathcal{T}_G(\beta(1))}(\alpha(1),\beta(1))$

(2) donne $(\gamma(\alpha\backslash 2),\beta\backslash 2) \in T_A^{\mathcal{T}_G(\beta(1))}(\alpha,\beta)$

donc d' après (1) $\gamma(\alpha\backslash 2) \equiv_G \beta\backslash 2$

d' où $\beta(1)\gamma(\alpha\backslash 2) \equiv_G \beta$ par stabilité de $\equiv_G$ pour "."

alors $\alpha(1)(\alpha\backslash 2) \equiv_G \beta(1)\gamma(\alpha\backslash 2)$ car $\alpha \equiv_G \beta$

par conséquent , $\alpha(1) \equiv_G \beta(1)\gamma$ car $\equiv_G$ est simplifiable pour "."

iii) est donc établie

iv) i) à iii) valident entièrement la proposition 5.2.2 $\square$

5.3 Canonicité des systèmes de $\prod(\mathbf{N})$

On se propose de montrer que l' opérateur S (cf. 2.5) est bien défini et est calculable c' est à dire

Proposition 5.3.1 : $\forall\, \mathbb{A} \in \prod(\mathbf{N})$, $\mathbb{A}$ est noethérien et confluent

Preuve : Considérons un élément $\mathbb{A}$ quelconque de $\prod(\mathbf{N})$ non vide

car si $\mathbb{A} = \emptyset$ alors la proposition est triviale

$$\text{donc } \forall\, (S,\alpha),\ (S',\alpha') \in \mathbb{A} \quad \left| \begin{array}{ll} \tau_G(S) = \tau_G(\alpha) & (D1) \\ \alpha \neq \alpha' \Rightarrow S \neq S' & (D2) \\ \alpha' \neq S & (D3) \end{array} \right.$$

soit $\gamma_{\mathbb{A}}$ l' endomorphisme sur $(\Sigma \cup \mathbf{N})^*$ défini sur $\Sigma \cup \mathbf{N}$ par

$$\forall\, U \in \Sigma \cup \mathbf{N},\quad \gamma_{\mathbb{A}}(U) = \left| \begin{array}{ll} \alpha & \text{si } (U,\alpha) \in \mathbb{A} \quad \text{a un sens d' après (D2)} \\ U & \text{sinon} \end{array} \right.$$

$\gamma_{\mathbb{A}}$ est appelée la règle de pleine substitution pour $\mathbb{A}$; elle possède les 4 propriétés suivantes

P1 $\quad \forall\, U \in (\Sigma \cup \mathbf{N})^*,\ U \overset{*}{\underset{A}{\to}} \gamma_{\mathbb{A}}(U)$

comme $\gamma_{\mathbb{A}}$ est un endomorphisme alors par induction sur $|U| \geqslant 0$, il suffit de montrer que P1 est vrai pour $U \in \Sigma \cup \mathbf{N}$

si $U \in \Sigma \cup (\mathbf{N}\text{-Ker}(\mathbb{A}))$ alors $\gamma_{\mathbb{A}}(U) = U$

si $U \in \text{Ker}(\mathbb{A})$ alors $(U, \gamma_{\mathbb{A}}(U)) \in \mathbb{A}$ donc $U \underset{A}{\to} \gamma_{\mathbb{A}}(U)$

d' où P1

P2 $\quad \forall\, U, V \in (\Sigma \cup \mathbf{N})^*,\ U \overset{*}{\underset{A}{\to}} V \Rightarrow \gamma_{\mathbb{A}}(U) \overset{*}{\underset{A}{\to}} \gamma_{\mathbb{A}}(V)$

comme $\overset{*}{\underset{A}{\to}}$ est transitive, une récurrence sur la longueur de la dérivation montre qu' il suffit d' établir P2 pour $U \underset{A}{\to} V$

or $U \underset{A}{\to} V \Leftrightarrow \exists\, X, Y \in (\Sigma \cup \mathbf{N})^*,\ \exists\, (S,\alpha) \in \mathbb{A} \,/\, U = XSY$ et $V = X\alpha Y$

donc $\gamma_{\mathbb{A}}(U) = \gamma_{\mathbb{A}}(X)\gamma_{\mathbb{A}}(S)\gamma_{\mathbb{A}}(Y) = \gamma_{\mathbb{A}}(X)\alpha\gamma_{\mathbb{A}}(Y)$

ainsi avec P1 , $\gamma_{\mathbb{A}}(U) \overset{*}{\underset{A}{\to}} \gamma_{\mathbb{A}}(X)\gamma_{\mathbb{A}}(\alpha)\gamma_{\mathbb{A}}(Y) = \gamma_{\mathbb{A}}(V)$

ce qui vérifie P2

P3 $\quad \forall\, U, V \in (\Sigma \cup \mathbf{N})^*,\ \forall\, n \geqslant 0,\ U \overset{n}{\underset{A}{\to}} V \Rightarrow V \overset{*}{\underset{A}{\to}} \gamma_{\mathbb{A}}^{\,n}(U)$

par récurrence sur $n \geqslant 0$

si $n = 0$ alors P3 est triviale

si $n = 1$ alors $U \underset{A}{\to} V$ et en reprenant les notations de P2 , on a

$\gamma_{\mathbb{A}}(U) = \gamma_{\mathbb{A}}(X)\alpha\gamma_{\mathbb{A}}(Y)$

d' où avec P1 , $V = X\alpha Y \overset{*}{\underset{A}{\to}} \gamma_{\mathbb{A}}(U)$

supposons P3 vrai pour $n \geqslant 1$ et montrons le pour n+1

$U \overset{n+1}{\underset{A}{\to}} V \Rightarrow \exists\, W \in (\Sigma \cup \mathbf{N})^* \,/\, U \overset{n}{\underset{A}{\to}} W$ et $W \underset{A}{\to} V$

donc d' après l' hypothèse de récurrence et le cas n = 1

$V \overset{*}{\underset{A}{\to}} \gamma_{\mathbb{A}}(W)$ et $W \overset{*}{\underset{A}{\to}} \gamma_{\mathbb{A}}^{\,n}(U)$

ainsi P2 et la transitivité de $\overset{*}{\underset{A}{\rightarrow}}$ donnent $V \overset{*}{\underset{A}{\rightarrow}} \gamma_{\mathbf{A}}^{n+1}(U)$

la récurrence est établie et P3 par la même occasion

P4 $\forall\, U \in (\Sigma \cup \mathbf{N})^*, \forall\, m, n \in \mathbb{N},\ m \leqslant n \Rightarrow \gamma_{\mathbf{A}}^{m}(U) \overset{*}{\underset{A}{\rightarrow}} \gamma_{\mathbf{A}}^{n}(U)$

en utilisant P1 et la transitivité de $\overset{*}{\underset{A}{\rightarrow}}$, une récurrence sur $n-m \geqslant 0$ prouve P4

i) $\forall\, U, V \in (\Sigma \cup \mathbf{N})^*,\ U \overset{*}{\underset{A}{\rightarrow}} V \Rightarrow \tau_G(U) = \tau_G(V)$ et $|V| \geqslant |U|$?

par induction sur la longueur de la dérivation

ii) $\forall\, S \in \mathrm{Ker}(A), \forall\, \alpha \in \mathbf{N}^*,\ S \overset{+}{\underset{A}{\rightarrow}} \alpha \Rightarrow \forall\, i \in [\,|\alpha|\,],\ \alpha(i) \neq S$?

$S \overset{+}{\underset{A}{\rightarrow}} \alpha$ donc d'après i) $|\alpha| \geqslant 1$ et $\tau_G(\alpha) = \tau_G(S)$

soit $|\alpha| = 1$ alors $\alpha \neq S$ d'après (D3)

soit $|\alpha| > 1$ alors $\tau_G(S) = \tau_G(\alpha) = \sum_{i=0}^{|\alpha|} \tau_G(\alpha(i))$

or G est $\perp$ libre et réduite donc $\forall\, i \in [\,|\alpha|\,],\ \tau_G(\alpha(i)) < \tau_G(S)$

i.e. $\forall\, i \in [\,|\alpha|\,],\ \alpha(i) \neq S$

ii) est donc établi

iii) $\forall\, U \in (\Sigma \cup \mathbf{N})^*,\ \gamma_{\mathbf{A}}^{\mathrm{Card}(\mathbf{A})}(U) \in (\Sigma \cup (\mathbf{N}-\mathrm{Ker}(\mathbf{A})))^*$?

par induction sur $\mathrm{Card}(\mathbf{A}) \geqslant 1$?

si $\mathrm{Card}(\mathbf{A}) = 1$ alors iii) provient de ii)

si iii) est vrai $\forall\, \mathbf{A} \in \prod(\mathbf{N})\ /\ \mathrm{Card}(\mathbf{A}) = n$ et $n \geqslant 1$

alors montrons iii) pour $\mathbf{A} \in \prod(\mathbf{N})$ et $\mathrm{Card}(\mathbf{A}) = n+1$

d'après P4 et ii) $\forall\, S \in (\Sigma \cup \mathbf{N})^*$,

$S \overset{*}{\underset{A}{\rightarrow}} \gamma_{\mathbf{A}}^{n+1}(S) = \gamma_{\mathbf{A}}^{n}(\gamma_{\mathbf{A}}(S)) \in (\Sigma \cup (\mathbf{N}-\{S\}))^*$ (1)

posons $\mathbf{A}' = \mathbf{A} - \{(S, \gamma_{\mathbf{A}}(S))\}$

par application de ii) $\gamma_{\mathbf{A}}^{n+1}(S) = \gamma_{\mathbf{A}'}^{n}(\gamma_{\mathbf{A}}(S))$

donc par hyp. d'induction, $\gamma_{\mathbf{A}}^{n+1}(S) \in (\Sigma \cup (\mathbf{N}-\mathrm{Ker}(\mathbf{A}')))^*$ (2)

(1) et (2) $\Rightarrow \forall\, S \in \mathrm{Ker}(\mathbf{A}),\ \gamma_{\mathbf{A}}^{n+1}(S) \in (\Sigma \cup (\mathbf{N}-\mathrm{Ker}(\mathbf{A})))^*$

or $\gamma_{\mathbf{A}}$ est un endomorphisme donc iii) est vrai pour n+1

l'induction est terminée

iii) est donc valide

iv) $\forall\, U, V \in (\Sigma \cup \mathbf{N})^*,\ U \overset{|U|\mathrm{Card}(\mathbf{A})+1}{\underset{A}{\longrightarrow}} V \Rightarrow |V| > |U|$?

par récurrence sur $|U| \geqslant 1$

le cas $|U| = 0$ est exclu vu qu'au moins une réécriture porte sur U

pour $|U| = 1$: de même, il faut que $U \in \mathrm{Ker}(\mathbf{A})$

il s'en suit que $|V| > |U|$

car sinon on peut extraire une suite de $\mathrm{Card}(\mathbf{A})+1$ éléments de $\mathbf{A}$

$(S_i)_{0 \leqslant i \leqslant \mathrm{Card}(\mathbf{A})}$ tels que

$S_0 = U,\ \forall\, i \in [\mathrm{Card}(\mathbf{A})]\ \ S_{i-1} \underset{A}{\rightarrow} S_i,\ \ S_{\mathrm{Card}(\mathbf{A})} \underset{A}{\rightarrow} V$

donc $\exists\ i, j \in \mathbb{N}\ /\ 0 \leq i < j \leq Card(\mathcal{A})$ et $S_j = S_i$

et comme $S_i \xrightarrow[\mathcal{A}]{+} S_j$, on obtient une contradiction d' après ii)

d' où iv) pour $|U| = 1$

supposons iv) vrai pour $|U| = n$ et $n \geq 1$ et montrons iv) pour $|U| = n+1$

soient $U' = U(1)$ et $U'' = U\backslash 2$ donc $U = U'U''$, $|U'| = 1$, $|U''| = n$

mais $\exists\ p$, $0 \leq p \leq |U|\,Card(\mathcal{A}) +1$, $\exists\ V', V'' \in (\Sigma \cup \mathbf{N})^* \ /\ U' \xrightarrow[\mathcal{A}]{p} V'$ et $U'' \xrightarrow[\mathcal{A}]{q} V''$ avec $V = V'V''$ et $q = 1+ |U|\,Card(\mathcal{A}) -p$

soit $p \geq Card(\mathcal{A}) +1$ alors d' après le cas précédent $|V'| > |U'|$

or d' après i) , $|V''| \geq |U''|$

d' où $|V| = |V'|+|V''| > |U'|+|U''| = |U|$

soit $p \leq Card(\mathcal{A})$ alors $q \geq |U''|\,Card(\mathcal{A}) +1$

donc $|V''| > |U''|$ par hypothèse de récurrence

et comme d' après i) $|V'| \geq |U'|$ alors $|V| > |U|$

ainsi iv) est vrai pour n+1

la récurrence est établie et iv) est démontré

v) $\mathcal{A}$ est noethérien ?

notons $k = Card(\mathcal{A}) +1$. Montrons tout d' abord que

$\forall\ U, V \in (\Sigma \cup \mathbf{N})^*$, $U \xrightarrow[\mathcal{A}]{*} V \Rightarrow V \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{k}(U)$ (I) ?

pour n la longueur de la dérivation de U en V

soit $n \leq k$

alors d' après P4 , $\gamma_{\mathcal{A}}^{n}(U) \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{k}(U)$ et d' après P3 , $V \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{n}(U)$

ce qui implique $V \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{k}(U)$

soit $n > k$

or d' après P3 , $V \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{n}(U)$; de plus d' après iii) , $\gamma_{\mathcal{A}}^{n}(U) = \gamma_{\mathcal{A}}^{k}(U)$

d' où $V \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{k}(U)$

(I) est donc établi

par conséquent , $\forall\ U \in (\Sigma \cup \mathbf{N})^*$, $\exists\ n_U \geq 0\ /\ Im(\{U\}, \xrightarrow[\mathcal{A}]{n_U}) = \emptyset$

car sinon iv) implique que $\exists\ n \geq 0$, $\exists\ V \in (\Sigma \cup \mathbf{N})^*\ /$

$U \xrightarrow[\mathcal{A}]{n} V$ et $|V| > |\gamma_{\mathcal{A}}^{k}(U)|$

or d' après (I) et i) $|V| \leq |\gamma_{\mathcal{A}}^{k}(U)|$ ce qui donne une contradiction

en définitive , $\mathcal{A}$ est noethérien

v) $\mathcal{A}$ est confluent ?

$\forall\ U \in (\Sigma \cup \mathbf{N})^*$, $\forall\ U', V' \in Im(\{U\}, \xrightarrow[\mathcal{A}]{*})$

$\exists\ m, n \in \mathbb{N}\ /\ U \xrightarrow[\mathcal{A}]{m} U'$ et $V \xrightarrow[\mathcal{A}]{n} V'$

donc d' après P3 $U' \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{m}(U)$ et $V' \xrightarrow[\mathcal{A}]{*} \gamma_{\mathcal{A}}^{n}(U)$

il en résulte avec P4 que $\gamma_{\mathcal{A}}^{max(m,n)}(U) \in Im(\{U'\}, \xrightarrow[\mathcal{A}]{*}) \cap Im(\{U'\}, \xrightarrow[\mathcal{A}]{*})$

la confluence de $\mathcal{A}$ est donc vérifiée

remarque :

la confluence de $\mathcal{A}$ pouvait être obtenu directement à l' aide du théoreme de Knuth-Bendix ($\mathcal{A}$ n' admettant pas de paire critique d' aprés D2)

v) et vi) ont établis entièrement la proposition □

5.4 Caractéristiques de l' algorithme

On se propose tout d' abord de montrer que l' algorithme est bien défini et qu' il construit toujours un arbre fini .

Proposition 5.4.1 : Soient $\alpha,\beta \in \mathbf{N}^*$

Soient $u_0 = \perp$, $A_0 = \emptyset$, $T_0 = \{(\perp,(\alpha,\beta))\}$ l' arbre réduit à (α,β)

alors $\mathrm{Const}(T_0, u_0, A_0)$ est un arbre fini (non gradué) sur $\mathbf{N}^* \times \mathbf{N}^*$

Preuve : On définit récursivement la suite $((T_i, u_i, A_i))_{i\geqslant 0}$, éventuellement infinie, de la façon suivante

Pour (T_i, u_i, A_i) défini

Si Const appliqué à (T_i,u_i,A_i) nécessite un nouvel appel à Const (Const est réentrant)

Alors les nouveaux arguments de Const sont $(T_{i+1},u_{i+1},A_{i+1})$

Sinon (T_i,u_i,A_i) est le dernier élément de la suite

ls

i) $\forall\ i \geqslant 0$, si A_i existe alors $A_i \in \prod(\mathbf{N})$?

par récurrence sur $i \geqslant 0$

pour $i = 0$: A_0 existe et $A_0 = \emptyset \in \prod(\mathbf{N})$

supposons que A_{i+1} existe et que $A_i \in \prod(\mathbf{N})$ alors montrons que $A_{i+1} \in \prod(\mathbf{N})$

soit $A_{i+1} = A_i \in \prod(\mathbf{N})$

soit $A_{i+1} \neq A_i$ on a donc appliqué une T_A transformation c' est à dire

pour $(\lambda',\mu') = S(T_i(u_i),A_i)$ (1)

on a $\min(|\lambda'|,|\mu'|) \leqslant 1$, $\mathcal{T}_G(\lambda') = \mathcal{T}_G(\mu')$, $\lambda' \neq \mu'$ (2)

$$A_{i+1} - A_i = \left|\begin{array}{ll} \{(\lambda',\mu')\} & \text{si } |\lambda'| = 1 \\ \{(\mu',\lambda')\} & \text{sinon} \end{array}\right. \quad (3)$$

(2) $\Rightarrow$ $\min(|\lambda'|,|\mu'|) = 1$

d' où d' après (3) l' élément de $A_{i+1} - A_i \in \mathbf{N} \times \mathbf{N}^+$

donc avec (1) et (3) cet élément $\in (\mathbf{N} - \mathrm{Ker}(A_i)) \times (\mathbf{N} - \mathrm{Ker}(A_i))^+$

de plus par hypothèse $A_i \in \prod(\mathbf{N})$ et que $\mathcal{T}_G(\lambda') = \mathcal{T}_G(\mu')$

alors $A_{i+1} \in \prod(\mathbf{N})$ par définition de $\prod(\mathbf{N})$

la récurrence est établie

d' où i)

remarque : i) signifie, à des vérifications près (laissées au lecteur), que

pour (T_i, u_i, A_i) connu, on sait décider si c' est le dernier élément de la suite et dans le cas contraire construire $(T_{i+1}, u_{i+1}, A_{i+1})$

ii) La suite $((T_i,u_i,A_i))_{i\geqslant 0}$ est finie ?

supposons que la suite $((T_i,u_i,A_i))_{i\geqslant 0}$ soit infinie (I)

a) montrons pour $n \in \mathbf{N}$, si $S(T_n(u_n),A_n) \not\in I_{\mathbf{N}^*}$ alors $\exists\ m > n \ /\ A_m \neq A_n$

notons $(\lambda,\mu) = S(T_n(u_n), A_n)$

donc $\forall\ j \in [\,|\lambda|\,]$, $\forall\ k \in [\,|\mu|\,]$ $\lambda(j)$, $\mu(k) \not\in \mathrm{Ker}(A_n)$ (1)

de plus $\lambda \neq \mu$ et $\tau_G(\lambda) = \tau_G(\mu)$

donc $\exists\ i \in [\,|\lambda|\,] \ /\ \lambda(i) \neq \mu(i)$ et $\forall\ j \in [i-1]$ $\lambda(j) = \mu(j)$

alors par définition de Const, i-1 T_B transformation(s) sont appliquée(s)

d' où $S(T_p(u_p),A_p) = (\lambda\backslash i,\mu\backslash i)$ et $A_p = A_n$ (2)

avec $p = m+i-1 \geqslant m$

soit $\min(|\lambda\backslash i|, |\mu\backslash i|) = 1$

alors, d' après (1), une T_A transformation est appliquée

d' où avec (1) et (2)

$$A_{p+1} - A_p = \left|\begin{array}{ll} \{(\lambda\backslash i,\mu\backslash i)\} & \text{si } |\lambda\backslash i| = 1 \\ \{(\mu\backslash i,\lambda\backslash i)\} & \text{sinon} \end{array}\right.$$

ainsi $m = p+1 > n$ convient pour a)

soit $|\lambda\backslash i|, |\mu\backslash i| \geqslant 2$

par symétrie, on peut supposer que $\tau_G(\lambda(i)) \geqslant \tau_G(\mu(i))$

d' après (1) (2) et (I), une T_B transformation est appliquée en u_p

et $\exists\ \eta \in \mathbf{N}^* \ /\ T_{p+1}(u_{p+1}) = (\lambda(i),\mu(i)\eta)$ avec $A_{p+1} = A_p$

ainsi $A_{p+1} = A_n$ ce qui donne avec (1)

$S(T_{p+1}(u_{p+1}), A_{p+1}) = (\lambda(i), \mu(i)S(\eta,A_n)) \not\in A_n$

or $\lambda(i) \neq \mu(i)$ donc avec (I) $A_{p+2} - A_n = \{S(T_{p+1}(u_{p+1}), A_{p+1})\}$

ainsi $m = p+2 > n$ convient pour a)

a) est donc vrai

b) or, d' après (I), $\forall\ p \geqslant 0, \exists\ n \geqslant p \ /\ S(T_n(u_n), A_n) \not\in I_{\mathbf{N}^*}$

d' où d' après a), $\forall\ p \geqslant 0, \exists\ m > p \ /\ A_m \neq A_p$

et comme $\forall\ i,j \in \mathbf{N}$, $i \leqslant j \Rightarrow A_i \subset A_j$

alors $\forall\ p \geqslant 0, \exists\ m > p \ /\ \mathrm{Card}(A_m) > \mathrm{Card}(A_p)$

par conséquent $\exists\ q > 0 \ /\ \mathrm{Card}(A_q) \geqslant \mathrm{Card}(\mathbf{N})$

ce qui donne une contradiction car d'après i) $A_q \in \prod(\mathbf{N})$

d' où (I) est faux et donc ii) est démontré

iii) on peut noter d' après ii), n l' indice du dernier élément de la suite $((T_i, u_i, A_i))_{i\geqslant 0}$

alors $\mathrm{Const}(T_n, u_n, A_n)$ est un arbre fini

de plus $\mathrm{Const}(T_0, u_0, A_0) = \mathrm{Const}(T_n, u_n, A_n)$

il en résulte que la proposition 5.4.1 est établie □

Ainsi, $\forall\ \alpha,\beta \in \mathbf{N}^*$, $\mathrm{Arbre}_G(\alpha,\beta)$ existe et est un arbre fini

La décidabilité de $\equiv_G$ est donnée par

Proposition 5.4.2 : $\forall\ \alpha,\beta \in \mathbf{N}^*,\ \alpha \equiv_G \beta \Leftrightarrow \forall\ u \in \mathrm{Fl}(T))\ \ T(u) \in I_{\mathbf{N}^*}$ avec $T = \mathrm{Arbre}_G(\alpha,\beta)$

On se propose d' établir deux lemmes avant de valider cette proposition

Lemme 1 : $\forall\ (\alpha,\beta) \in \mathbf{N}^* \times \mathbf{N}^*,\ \forall\ A$ système canonique $\subset \equiv_G$

$$\alpha \equiv_G \beta \Leftrightarrow S((\alpha,\beta),A) \subset \equiv_G$$

On dit alors que S est valide

Preuve du lemme 1 :

Comme $A \subset \equiv_G$ et $\equiv_G$ est compatible pour ".", est symétrique et transitive

alors $\overset{*}{\underset{A}{\rightarrow}} \subset \equiv_G$

donc $\forall\ \alpha,\ \beta \in \mathbf{N}^*,\ \alpha \equiv_G \alpha{\downarrow}A$ et $\beta \equiv_G \beta{\downarrow}A$

d' où $\alpha \equiv_G \beta \Leftrightarrow \alpha{\downarrow}A \equiv_G \beta{\downarrow}A$ □

Lemme 2 : Soit A une relation non vide binaire sur $\mathbf{N}^*$

$\forall\ (\alpha,\beta) \in \overset{*}{\underset{A}{\rightarrow}},\ \exists\ (\alpha',\beta') \in A\ /\ \mathrm{Div}(\alpha',\beta') \leqslant \mathrm{Div}(\alpha,\beta)$

Preuve du lemme 2 : par induction sur la longueur de la dérivation

Si $\alpha \overset{0}{\underset{A}{\rightarrow}} \beta$ i.e. $\alpha = \beta$ alors $\mathrm{Div}(\alpha,\beta) = \infty$

le lemme est alors vrai car A est non vide

Si $\alpha \underset{A}{\rightarrow} \beta$ alors $\exists\ \lambda,\ \mu \in \mathbf{N}^*,\ \exists\ (\alpha',\beta') \in A\ /\ \alpha = \lambda\alpha'\mu$ et $\beta = \lambda\beta'\mu$

donc d' après la proposition 5.1.2 , $\mathrm{Div}(\alpha,\beta) = \mathrm{Div}(\alpha',\beta') + \tau_G(\lambda) + \tau_G(\mu)$

ainsi $\mathrm{Div}(\alpha',\beta') \leqslant \mathrm{Div}(\alpha,\beta)$ avec $(\alpha',\beta') \in A$

Supposons le lemme vrai $\forall\ (\alpha,\beta) \in \overset{m}{\underset{A}{\rightarrow}}$ avec $0 \leqslant m \leqslant n$ et $n \geqslant 1$

et montrons le pour $(\alpha,\beta) \in \overset{n+1}{\underset{A}{\rightarrow}}$

donc $\exists\ \gamma \in \mathbf{N}^*\ /\ \alpha \underset{A}{\rightarrow} \gamma$ et $\gamma \overset{n}{\underset{A}{\rightarrow}} \beta$

le cas précédent et l' hypothèse d' induction nous autorisent à dire que

$\exists\ (\alpha',\gamma'),\ (\gamma'',\beta') \in A\ /\ \mathrm{Div}(\alpha',\gamma') \leqslant \mathrm{Div}(\alpha,\gamma)$ et $\mathrm{Div}(\gamma'',\beta') \leqslant \mathrm{Div}(\gamma,\beta)$

d' où d' après la proposition 5.1.1 $\min(\mathrm{Div}(\alpha',\gamma'),\mathrm{Div}(\gamma'',\beta')) \leqslant \mathrm{Div}(\alpha,\beta)$

L' induction est donc terminée et le lemme 2 démontré □

Preuve de la proposition 5.4.2 :

Soient $\alpha,\beta \in \mathbf{N}^*$. Notons $T = \mathrm{Arbre}_G(\alpha,\beta)$ et A le système de réécriture (final) obtenu

i) Condition nécessaire : $\alpha \equiv_G \beta \Rightarrow \forall\ u \in \mathrm{Fl}(T),\ T(u) \in I_{\mathbf{N}^*}$?

on montre, de façon immédiate, que $\forall\ u \in \mathrm{Dom}(T),\ T(u) \in \equiv_G$

par induction sur $|u| \geqslant 0$, en utilisant la condition nécessaire de la validité de T_A, T_B et S (proposition 5.2.2 et lemme 1)

ii) Condition suffisante : $(\alpha \not\equiv_G \beta)$ et $(\forall\ u \in \mathrm{Fl}(T),\ T(u) \in I_{\mathbf{N}^*})$ sont contradictoires ?

a) $A - \equiv_G \neq \emptyset$

car sinon $A \subset \equiv_G$ et comme $\forall\, u \in Fl(T)$, $T(u) \in \equiv_G$

alors de façon analogue à i), par induction sur $d(T) - |u| \geqslant 0$,

en utilisant la condition suffisante de la validité de T_A, T_B, S

on obtient $\forall\, u \in Dom(T)$, $T(u) \in \equiv_G$

donc $T(\perp) = (\alpha,\beta) \in \equiv_G$ exclu par hypothèse d' où a)

b) $\forall\, (\lambda,\mu) \in A - \equiv_G$, $\exists\, (\lambda',\mu') \in A\ /\ Div(\lambda',\mu') < Div(\lambda,\mu)$?

b1) $(\lambda,\mu) \in A$ donc

$$u = \min_{\leqslant_{lex}} \{v \in Dom(T)\ /\ T(v) \in \{(\lambda,\mu),(\mu,\lambda)\}\}\ \text{existe}$$

alors, d' après l' algorithme, une T_A transformation a été appliquée en u

d' où $\emptyset \neq T_A(u) = (\lambda,\mu) \notin \equiv_G$

et comme T_A est monotone (proposition 5.2.1) alors

$$\exists\, i \in \mathbb{N}_+\ /\ ui \in Dom(T) \text{ et } Div(T(ui)) < Div(\lambda,\mu) < \infty$$

b2) Montrons $\forall\, u \in Dom(T)\ /\ T(u) \notin \equiv_G$,

$$\exists\, (\lambda',\mu') \in A\ /\ Div(\lambda',\mu') \leqslant Div(T(u))\ ?$$

par induction sur $d(T\backslash u) \geqslant 1$ ($T\backslash u$ désigne le sous-arbre de T en u)

$d(T\backslash u) = 0$ est exclu car $\forall\, v \in Fl(T)$, $T(v) \in I_{\mathbb{N}^*}$ et $T(u) \notin \equiv_G$

pour $d(T\backslash u) = 1$: comme $T(u) \notin \equiv_G$ et $Im(Fl(T),T) \subset I_{\mathbb{N}^*}$

et que de plus T_A et T_B sont des transformations valides

alors seul l' opérateur de simplification S est applicable en u

et donc $T(u) = (\lambda,\mu)$ se simplifie en $(\lambda \downarrow A, \mu \downarrow A)$ avec $\mu \downarrow A = \lambda \downarrow A$

d' où d' après la proposition 5.1.1

$$\min(Div(\lambda,\lambda \downarrow A), Div(\mu,\mu \downarrow A)) \leqslant Div(\lambda,\mu) < \infty$$

par symétrie de λ et μ, on peut supposer que

$$Div(\lambda,\lambda \downarrow A) \leqslant Div(\lambda,\mu) < \infty$$

par conséquent d' après le lemme 2 b2) est vrai si $d(T\backslash u) = 1$ avec $T(u) \notin \equiv_G$

supposons b2) vrai $\forall\, u \in Dom(T)\ /\ T(u) \notin \equiv_G$ et $d(T\backslash u) \leqslant n$

pour $n \geqslant 1$ et soit $u \in Dom(T)\ /\ T(u) \notin \equiv_G$ et $d(T\backslash u) = n+1$

alors montrons $\exists\, (\lambda',\mu') \in A\ /\ Div(\lambda',\mu') \leqslant Div(T(u))$? (I)

soit u est développé par une T_A ou T_B transformation, monotone

$\Rightarrow\ \exists\, i \in \mathbb{N}_+\ /\ ui \in Dom(T)$ et $Div(T(ui)) < Div(T(u)) < \infty$ (1)

alors $T(ui) \notin \equiv_G$ et $d(T(ui)) \leqslant n$

l' hypothèse de récurrence est applicable à $T(ui)$

il s' en suit que $\exists\, (\lambda',\mu') \in A\ /\ Div(\lambda',\mu') \leqslant Div(T(ui))$ (2)

(1) et (2) donnent (I)

soit u est développé par S (relativement à un système dont les éléments sont des labels dont les noeuds précèdent u par ordre lexicographique)

alors pour $T(u) = (\lambda,\mu)$, $T(u1) = (\lambda \downarrow A, \mu \downarrow A)$

si $Div(\lambda \downarrow A, \mu \downarrow A) \leqslant Div(\lambda,\mu)$

par hypothèse d' induction , on obtient (I)

si $Div(\lambda \downarrow A, \mu \downarrow A) > Div(\lambda,\mu)$ (1)

alors d' après la proposition 5.1.1 , on a

$$\min (Div(\lambda \downarrow A, \mu \downarrow A), Div(\lambda \downarrow A, \lambda), Div(\mu \downarrow A, \mu)) \leqslant Div(\lambda,\mu)$$

ce qui donne avec (1)

$$\min (Div(\lambda \downarrow A, \lambda), Div(\mu \downarrow A, \mu)) \leqslant Div(\lambda,\mu) < \infty$$

par symétrie de λ et μ , on peut supposer que

$$Div(\lambda, \lambda \downarrow A) \leqslant Div(\lambda,\mu) < \infty$$

d' où d' après le lemme 2 , on obtient le $(\lambda',\mu') \in A$ satisfaisant (I)

(I) est donc toujours vérifié

l' induction est terminée et b2) est donc démontré

b1) et b2) établissent immédiatement b)

c) a) et b) $\Rightarrow \exists\ (\lambda,\mu) \in A\ /\ Div(\lambda,\mu) \leqslant 0$

on a aboutit à une contradiction et ii) est donc établi

iii) i) et ii) ont démontré la proposition 5.4.2 □

6 Validité du critère

La validation s' effectue en trois étapes

tout d' abord, on se dote d' un outil d' évaluation, relation binaire sur $P(\mathbf{N}^*\times\mathbf{N}^*)$, dit de comparaison par T_A transformation

ensuite, on montre à l' aide de trois caractéristiques, combien coûte l' application de chacun des trois opérateurs T_A , T_B , S de construction de l' arbre d' équivalence

enfin, on évalue l' arbre d' équivalence, c' est à dire on le compare à celui obtenu uniquement par T_A transformation (arbre de transformation parallèle gauche)

6.1 Outil d' évaluation

On définit la relation binaire $\gg_G$ sur $P(\mathbf{N}^*\times\mathbf{N}^*)$ par

$$\forall A, B \subset \mathbf{N}^*\times\mathbf{N}^*,\ A \gg_G B \Leftrightarrow \text{si } \exists\, n \in \mathbf{N} \,/\, T_A^n(A) = \emptyset \text{ alors } T_A^n(B) = \emptyset$$

On dit alors que A est comparable par T_A transformation à B

On rappelle que T_A^n a été défini en 2.4

Le lecteur s' assurera que

$$\forall \alpha,\beta \in \mathbf{N}^*,\ \mathrm{Divg}(\alpha,\beta) = \left|\begin{array}{ll} \infty & \text{si } \alpha \equiv_G \beta \\ \inf\{n / T_A^n(\alpha,\beta) = \emptyset\} & \text{sinon} \end{array}\right.$$

en déduira que

P0 $\forall A, B \in P(\mathbf{N}^*\times\mathbf{N}^*) - \{\emptyset\},\ A \gg_G B \Leftrightarrow \min_{\Phi\in B}(\mathrm{Divg}\ \Phi) \leqslant \min_{\Phi\in A}(\mathrm{Divg}\ \Phi)$

et vérifiera que $\gg_G$ possède les propriétés élémentaires suivantes

P1 $\gg_G$ est un pré-ordre i.e. est réflexive et transitive

P2 $(\subset_{P(\mathbf{N}^*\times\mathbf{N}^*)-\{\emptyset\}}) \subset (\gg_G)$ et même

$$\forall A,B \subset \mathbf{N}^*\times\mathbf{N}^* \text{ et } A\neq\emptyset \text{ si } B\neq\emptyset,\ A \subset B \Rightarrow A \gg_G B$$

P3 $\gg_G$ est régulière pour $\cup$ i.e.

$$\forall A, B, C \subset \mathbf{N}^*\times\mathbf{N}^*,\ A \gg_G B \Rightarrow (A \cup C) \gg_G (B \cup C)$$

P4 $\gg_G$ est stable pour $\cup$ i.e.

$$\forall A, B, A', B' \subset \mathbf{N}^*\times\mathbf{N}^*,\ \text{si } A \gg_G B \text{ et } A' \gg_G B' \text{ alors } A \cup A' \gg_G B \cup B'$$

P5 $(P(\equiv_G) \times P(\mathbf{N}^*\times\mathbf{N}^*)) \subset (\gg_G)$ par condition nécessaire de la validité de T_A

Remarque : si P1 est vraie alors P3 $\Leftrightarrow$ P4

6.2 Evaluation des opérateurs

Pour A comparable par T_A transformation à B , évaluer un opérateur consiste à rechercher le nombre de T_A transformation(s) qu' il faut appliquer à B pour que l' application de l' opérateur à A lui soit comparable (par T_A transformation) .

On étend la valuation τ_G sur $\mathbf{N}^* \times \mathbf{N}^*$ par

$$\forall\ \alpha,\beta \in \mathbf{N}^*,\quad \tau_G(\alpha,\beta) = \min\ \{\tau_G(\alpha), \tau_G(\beta)\}$$

Caractéristique 1 : évaluation de T_A

Soient $A, B \in P(\mathbf{N}^* \times \mathbf{N}^*)$. Soient $m,n \in \mathbf{N}$ / $m \leqslant n$

$A \gg_G B \Rightarrow T_A^m(A) \gg_G T_A^n(B)$

Preuve :

soit $\forall\ i \in \mathbf{N},\ T_A^i(T_A^m(A)) \neq \emptyset$

donc $T_A^m(A) \subset \equiv_G$

d' où la caractéristique est vraie d' après P5

soit $\exists\ i \geqslant 0$ / $T_A^i(T_A^m(A)) = \emptyset$ i.e. $T_A^{i+m}(A) = \emptyset$

or $i+m \leqslant i+n$ et $T_A(\emptyset) = \emptyset$ d' où $T_A^{i+n}(A) = \emptyset$

mais $A \gg_G B$ donc $T_A^{i+n}(B) = \emptyset$ i.e. $T_A^i(T_A^n(B)) = \emptyset$

la caractéristique 1 est donc prouvée □

Caractéristique 2 : évaluation de T_B

soit $\Phi = (\alpha,\beta) \in \mathrm{Ker}(T_B) = \mathbf{N} . \mathbf{N}^+ \times \mathbf{N} . \mathbf{N}^+$ / $\tau_G(\alpha) = \tau_G(\beta)$

alors $\forall\ \Psi \in T_B(\alpha,\beta),\ 1 \leqslant \tau_G(\Psi) \leqslant \tau_G(\Phi)$ et $\{\Psi\} \gg_G T_A^{\tau_G(\Phi)-\tau_G(\Psi)}(\Phi)$

Preuve :

soit $T_B(\alpha,\beta) = \emptyset$ auquel cas la caractéristique est triviale

soit $T_B(\alpha,\beta) \neq \emptyset$

on peut supposer, par symétrie de α et β, que $\tau_G(\alpha(1)) \geqslant \tau_G(\beta(1))$

donc, par définition de T_B, on a

$$\exists\ \gamma \in \mathbf{N}^* \ / \ (\gamma,\perp) \in T_A^{\tau_G(\beta(1))}(\alpha(1),\beta(1))$$
$$\text{et} \qquad T_B(\alpha,\beta) = \{\ (\alpha(1),\beta(1)\gamma)\ ,\ (\gamma(\alpha\backslash 2),\beta\backslash 2)\ \}$$

donc $1 \leqslant \tau_G(\alpha(1),\beta(1)\gamma) \leqslant \tau_G(\alpha(1)) \leqslant \tau_G(\alpha) = \tau_G(\Phi)$

$1 \leqslant \tau_G(\gamma(\alpha\backslash 2),\beta\backslash 2) \leqslant \tau_G(\beta\backslash 2) \leqslant \tau_G(\beta) = \tau_G(\Phi)$

d' après P0 , il nous reste à montrer que

$$\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\gamma(\alpha\backslash 2),\beta\backslash 2) + \tau_G(\Phi) - \tau_G(\gamma(\alpha\backslash 2),\beta\backslash 2) \quad (\mathrm{I}\ 1)$$

et $$\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\alpha(1),\beta(1)\gamma) + \tau_G(\Phi) - \tau_G(\alpha(1),\beta(1)\gamma) \quad (\mathrm{I}\ 2)$$

i) $(\gamma,\perp) \in T_A^{\tau_G(\beta(1))}(\alpha(1),\beta(1))$ et $\tau_G(\beta(1)) \leqslant \tau_G(\alpha(1))$ donnent

$(\gamma(\alpha\backslash 2),\beta\backslash 2) \in T_A^{\tau_G(\beta(1))}(\alpha,\beta)$

par conséquent $\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\gamma(\alpha\backslash 2),\beta\backslash 2) + \tau_G(\beta(1))$ (1)

et comme $\tau_G(\beta(1)) = \tau_G(\beta) - \tau_G(\beta\backslash 2) = \tau_G(\Phi) - \tau_G(\beta\backslash 2)$

on obtient (I 1)

ii) d' après la proposition 5.1.4 , on a

$$\mathrm{Divg}(\alpha,\beta(1)\gamma(\alpha\backslash 2)) \leqslant \mathrm{Divg}(\alpha(1),\beta(1)\gamma) + \tau_G(\alpha\backslash 2) \quad (2)$$

et $$\mathrm{Divg}(\beta(1)\gamma(\alpha\backslash 2),\beta) = \mathrm{Divg}(\gamma(\alpha\backslash 2),\beta\backslash 2) + \tau_G(\beta(1)) \quad (3)$$

(1) et (3) $\Rightarrow$ $\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\beta(1)\gamma(\alpha\backslash 2),\beta)$

d' où $\mathrm{Divg}(\alpha,\beta) = \min(\mathrm{Divg}(\alpha,\beta),\mathrm{Divg}((\beta(1)\gamma(\alpha\backslash 2),\beta)$

ainsi d' après la proposition 5.1.5 , $\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\alpha,\beta(1)\gamma(\alpha\backslash 2))$

ce qui donne avec (2) $\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\alpha(1),\beta(1)\gamma) + \tau_G(\alpha\backslash 2)$

et comme $\tau_G(\alpha\backslash 2) = \tau_G(\alpha) - \tau_G(\alpha(1)) = \tau_G(\Phi) - \tau_G(\alpha(1))$

(I 2) est validé

iii) i) et ii) établissent entièrement la caractéristique 2 □

Caractéristique 3 :

soient $\Phi \in \mathbf{N}^*\times\mathbf{N}^*$ et $A \in \prod(\mathbf{N})$

alors $\{S(\Phi,A)\} \gg_G A \cup \{\Phi\}$

Preuve : i) On se propose d' établir un résultat plus général, à savoir

soit A une relation binaire quelconque sur $\mathbf{N}^*$

on étend $\xrightarrow[A]{}$ sur $\mathbf{N}^*\times\mathbf{N}^*$ par

$\forall\ (\alpha,\beta),\ (\alpha',\beta') \in \mathbf{N}^*\times\mathbf{N}^*$,

$(\alpha,\beta) \xrightarrow[A]{} (\alpha',\beta')$ si et seulement si

$(\alpha \xrightarrow[A]{} \alpha'$ et $\beta = \beta')$ oux $(\alpha = \alpha'$ et $\beta \xrightarrow[A]{} \beta')$

montrons alors que

$$\forall\ \Phi,\ \Phi' \in \mathbf{N}^*\times\mathbf{N}^*,\ \Phi \xleftrightarrow[A]{*} \Phi' \Rightarrow \{\Phi'\} \gg_G \{\Phi\} \cup A \quad (\mathrm{I})\ ?$$

par induction sur la longueur de la dérivation

si $\Phi \xleftrightarrow[A]{0} \Phi'$ i.e. $\Phi = \Phi'$ alors $\{\Phi'\} \subset \{\Phi\} \cup A$

d' où (I) d' après P2

si $\Phi = (\alpha,\beta) \xleftrightarrow[A]{} \Phi' = (\alpha',\beta')$ alors par symétrie de α et β ,

on peut supposer que $\alpha \xleftrightarrow[A]{} \alpha'$ et $\beta = \beta'$

or $\alpha \xleftrightarrow[A]{} \alpha' \Leftrightarrow \exists\ \lambda,\mu \in \mathbf{N}^*$ et $(\gamma,\gamma') \in A \cup A^{-1}$ /

$\alpha = \lambda\gamma\mu$ et $\alpha' = \lambda\gamma'\mu$

donc d' après la proposition 5.1.4

$$\begin{aligned}\mathrm{Divg}(\alpha,\alpha') &= \mathrm{Divg}(\gamma\mu,\gamma'\mu) + \tau_G(\lambda)\\ &\geqslant \mathrm{Divg}(\gamma\mu,\gamma'\mu) \geqslant \mathrm{Divg}(\gamma,\gamma')\end{aligned}$$

ainsi $\min(\mathrm{Divg}(\alpha,\beta),\mathrm{Divg}(\gamma,\gamma')) \leqslant \min(\mathrm{Divg}(\alpha,\beta),\mathrm{Divg}(\alpha,\alpha'))$

d' après 5.1.5 et $\beta = \beta'$ il vient

$\min(\mathrm{Divg}(\alpha,\beta),\mathrm{Divg}(\gamma,\gamma')) \leqslant \mathrm{Divg}(\alpha',\beta) = \mathrm{Divg}(\alpha',\beta')$

d' où d' après P0 $\{\Phi'\} \gg_G \{\Phi\} \cup A$

si (I) est vrai $\forall\ \Phi,\ \Phi' \in \mathbf{N}^*\times\mathbf{N}^*\ /\ \Phi \xleftrightarrow[A]{n} \Phi'$ et $n \geqslant 1$

alors montrons (I) pour $\Phi,\ \Phi' \in \mathbf{N}^*\times\mathbf{N}^*$ avec $\Phi \xleftrightarrow[A]{n+1} \Phi'$

donc $\exists\ \Phi'' \in \mathbf{N}^*\times\mathbf{N}^*\ /\ \Phi \xleftrightarrow[A]{n} \Phi''$ et $\Phi'' \xleftrightarrow[A]{} \Phi'$

par hypothèse d' induction et d' après le cas précédent, on obtient

$$\{\Phi''\} \gg_G \{\Phi\} \cup A \quad \text{et} \quad \{\Phi'\} \gg_G \{\Phi''\} \cup A$$

P1 et P3 donnent $\{\Phi'\} \gg_G \{\Phi\} \cup A$ c' est à dire (I)

l' induction est terminée et (I) est donc vraie

ii) pour $A \in \prod(\mathbf{N})$, $\forall \Phi \in \mathbf{N}^* \times \mathbf{N}^*$, $\Phi \xrightarrow[A]{*} S(\Phi,A)$

donc d' après i) $S(\Phi,A) \gg_G \{\Phi\} \cup A$

par conséquent, la caractéristique 3 est validée □

6.3 Evaluation de l' arbre d' équivalence

Pour T un arbre d' équivalence et $A \in \prod(\mathbf{N})$ la théorie (finale) associée , on note

$\text{Noeud}(A) = \{u \in \text{Dom}(T) / T(u) \in A \cup A^{-1}$ et $\forall v <_{lex} u$, $T(v) \notin \{T(u)\} \cup \{T(u)\}^{-1} \}$

ainsi, Noeud(A) est l' ensemble des noeuds de T qui ont été développés par T_A transformation

et pour $u \in \text{Dom}(T)$

$A_u = \{T(v) \in A / v \leqslant_{lex} u\} = \{T(v) / v \leqslant_{lex} u$ et $v \in \text{Noeud}(A)\}$

$D_u = \{uv \in \text{Dom}(T) / \forall w \in FG(v) - \{\perp, v\}$, $uw \notin \text{Noeud}(A)\}$

D_u est donc l' ensemble des noeuds de T que l' on peut obtenir par développement à partir de u jusqu' à ce que l' on rencontre des feuilles ou des noeuds de A

Préalablement, on présente deux lemmes d' évaluation de sous-arbres partiels de T

L' évaluation de tout sous-arbre partiel de T dont les noeuds ont été obtenus par T_B et S opérations, est donnée par

lemme 1 : Soit $u \in \text{Dom}(T) - \text{Noeud}(A)$

$$\forall v \in D_u , \ \mathcal{T}_G(T(v)) \leqslant \mathcal{T}_G(T(u)) \ \text{et} \ \{T(v)\} \gg_G T_A^{\mathcal{T}_G(T(u)) - \mathcal{T}_G(T(v))}(A_u \cup \{T(u)\})$$

Preuve : Par induction sur $|v| - |u| \geqslant 0$ où $u \in \text{Dom}(T) - \text{Noeud}(A)$ et $v \in D_u$

si $|v| - |u| = 0$ alors $u = v$ car $u \in FG(v)$

le lemme 1 est alors évident

si $|v| - |u| = 1$

comme $u \notin \text{Noeud}(A)$, v est obtenu à partir de u soit par simplification à l' aide de A_u , soit par T_B transformation , c' est à dire

soit $T(v) = S(T(u), A_u)$

donc $\mathcal{T}_G(T(v)) = \mathcal{T}_G(T(u))$ car $\forall (\lambda,\mu) \in A$, $\mathcal{T}_G(\lambda) = \mathcal{T}_G(\mu)$

et d' après la caractéristique 3 , $\{T(v)\} = \{S(T(u),A_u)\} \gg_G A_u \cup \{T(u)\}$

soit $T(v) \in T_B(T(u))$

donc d' après la caractéristique 2 , $\mathcal{T}_G(T(v)) \leqslant \mathcal{T}_G(T(u))$ et

$$\{T(v)\} \gg_G T_A^{\mathcal{T}_G(T(u)) - \mathcal{T}_G(T(v))}(T(u))$$

or $\{T(u)\} \gg_G A_u \cup \{T(u)\}$ d' après P2

d' où le lemme 1 d' après la caractéristique 1 et P1

le lemme est donc vrai pour $|v|-|u| = 1$

supposons que le lemme soit vrai

$\forall$ u $\in$ Dom(T)-Noeud(A) , $\forall$ v $\in D_u$ / $|v|-|u| = n$ et $n \geqslant 1$

alors montrons le pour u $\in$ Dom(T)-Noeud(A) et v $\in D_u$ / $|v|-|u| = n+1$

comme $|v|-|u| \geqslant 2$ et $v \in D_u$ on a $v = uiv'$ avec $i \in \mathbb{N}_+$ et $v' \in \mathbb{N}_+^*$

il s' en suit que $v \in D_{ui}$ et ui $\notin$ Noeud(A)

l' hypothèse d' induction est applicable à $|v|-|ui| = n$ d' où

$$\tau_G(T(v)) \leqslant \tau_G(T(ui)) \quad \text{et} \quad \{T(v)\} \gg_G T_A^{\tau_G(T(ui))-\tau_G(T(v))}(A_{ui} \cup \{T(ui)\}) \qquad (1)$$

de plus ui $\in D_u$ - Noeud(A) et $|ui|-|u| = 1$ donc en appliquant le cas précédent

$$\tau_G(T(ui)) \leqslant \tau_G(T(u)) \quad \text{et} \quad \{T(ui)\} \gg_G T_A^{\tau_G(T(u))-\tau_G(T(ui))}(A_u \cup \{T(u)\}) \qquad (2)$$

or $A_{ui} = A_u$ car ui $\notin$ Noeud(A) et $\tau_G(T(v)) \leqslant \tau_G(T(u))$ d' après (1) et (2)

donc (2) , P3 , P2 $\Rightarrow$ $A_{ui} \cup \{T(ui)\} \gg_G T_A^{\tau_G(T(u))-\tau_G(T(ui))}(A_u \cup \{T(u)\})$ (3)

d' où (1) , (3) , la caractéristique 1 et la transitivité de $\gg_G$ donnent le lemme

l' induction est terminée et par la même occasion le lemme 1 démontré $\square$

On peut évaluer maintenant tout sous-arbre partiel de T dont aucun noeud interne (non-terminal), autre que la raçine, est noeud de A à l' aide de

lemme 2 : Soit u $\in$ Noeud(A)

$\forall$ v $\in D_u$, $\tau_G(T(v)) \leqslant e + 1$ et $\{T(v)\} \gg_G T_A^{e+1-\tau_G(T(v))}(A_u)$

Preuve : $\forall$ u $\in$ Dom(T) ,

$D_u = \{u\} \cup \{ui \in \text{Noeud(A)} / i \in \mathbb{N}_+\} \cup \{v \in D_{ui} / i \in \mathbb{N}_+ \text{ et } ui \in \text{Dom(T)-Noeud(A)}\}$

donc pour $v \in D_u$, on a les 3 cas suivants

soit v = u

or u $\in$ Noeud(A) donc $\tau_G(T(v)) \leqslant \ell \leqslant e+1$

de plus $T(u) \in A_u \cup A_u^{-1}$

donc d' après P2 et par symétrie de T_A , $\{T(u)\} \gg_G A_u$

d' où le lemme d' après la caractéristique 1 et la transitivité de $\gg_G$

soit v = ui $\in$ Noeud(A) avec i $\in \mathbb{N}_+$

u $\in$ Noeud(A) donc v est obtenu à partir de u par T_A transformation donc

$\tau_G(T(v)) \leqslant e$ et $T(v) \in T_A(T(u))$

d' où le lemme d' après la caractéristique 1 et la transitivité de $\gg_G$

soit v $\in D_{ui}$ et ui $\in$ Dom(T) - Noeud(A)

alors par application du lemme 1 , on a $\tau_G(T(v)) \leqslant \tau_G(T(ui))$

et $\{T(v)\} \gg_G^{\tau_G(T(ui))-\tau_G(T(v))} T_A(A_{ui} \cup \{T(ui)\})$ (1)

comme $u \in Noeud(A)$ donc $\tau_G(T(ui)) \leqslant e$ (2)

d' où $e+1-\tau_G(T(v)) \geqslant 1$

de plus $T(ui) \in T_A(T(u))$ car $u \in Noeud(A)$

et $A_{ui} = A_u$ car $ui \notin Noeud(A)$

il s' en suit que $A_{ui} \cup \{T(ui)\} \gg_G A_u \cup T_A(T(u)) \gg_G T_A(A_u)$

d' où le lemme en utilisant (1) (2), la caractéristique 1 et la transitivité de $\gg_G$

dans tous les cas, le lemme 2 est vérifié □

Proposition 6.3.1 :

Soient $\alpha, \beta \in \mathbb{N}^* \ / \ \min(|\alpha|, |\beta|) = 1$

Notons $T = Arbre_G(\alpha,\beta)$ et A le système de réécriture (final) obtenu

alors $\forall u \in Dom(T)$, $\{T(u)\} \gg_G^{\max(0,\, eCard(A) + 1-\tau_G(T(u)))} T_A(\alpha,\beta)$

Preuve : si $Dom(T)$ est réduit à $\{\perp\}$ alors la proposition est triviale

si $Dom(T) \neq \{\perp\}$

alors $\perp \in Noeud(A)$

Notons $\forall i \in [Card(A)]$, u_i le $i^{ème}$ élément de $Noeud(A)$ pour l' ordre lexicographique

et montrons par induction sur $i \in [Card(A)]$ que

$\forall u \in D_{u_i}$, $\{T(u)\} \gg_G^{ei + 1-\tau_G(T(u))} T_A(\alpha,\beta)$ (I) ?

pour $i = 1$, $u_1 = \perp$ et $A_\perp = \{(\alpha,\beta)\}$ (ou $\{(\beta,\alpha)\}$ si $1 = |\beta| < |\alpha|$)

donc d' après le lemme 2, $\forall u \in D_{u_1}$, $\{T(u)\} \gg_G^{e+1-\tau_G(T(u))} T_A(\alpha,\beta)$

c' est à dire (I) pour $i = 1$

si (I) est vrai $\forall j \in [i]$ avec $1 \leqslant i < Card(A)$

alors montrons que (I) est vrai pour i+1

d' après le lemme 2, $\forall u \in D_{u_{i+1}}$, $\{T(u)\} \gg_G^{e-\tau_G(T(u))+1} T_A(A_{u_{i+1}})$ (1)

or $\forall v \in A_{u_{i+1}}$, $\exists j \in [i] \ / \ v \in D_{u_j}$

donc par hypothèse d' induction

$\forall v \in A_{u_{i+1}}, \exists j \in [i] \ / \ \{T(v)\} \gg_G^{ej + 1-\tau_G(T(v))} T_A(\alpha,\beta)$

mais $\forall v \in Noeud(A)$, $\tau_G(T(v)) \geqslant 1$

d' où d' après la caractéristique 1, $A_{u_{i+1}} \gg_G^{ei} T_A(\alpha,\beta)$ (2)

d' où (1), (2) donnent (I) pour i+1

l' induction est terminée et (I) est établi

comme $Dom(T) = \bigcup_{u \in Noeud(A)} D_u$ on en déduit finalement la proposition 6.3.1 □

On peut maintenant établir une première partie du critère d' équivalence

Proposition 6.3.2 Soients $\alpha, \beta \in \mathbf{N}^* \ / \ \alpha \not\equiv_G \beta$

Si $\tau_G(\alpha) \neq \tau_G(\beta)$ alors $Divg(\alpha,\beta) \leqslant \min(\tau_G(\alpha),\tau_G(\beta)) + 1$

Si $\tau_G(\alpha) = \tau_G(\beta)$ et $\min(|\alpha|,|\beta|) = 1$ alors

$$Divg(\alpha,\beta) \leqslant \left| \begin{array}{l} e + 2 \text{ si } |G| = 2 \text{ et } e \neq 0 \\ (|G|-1)e + 1 \text{ si } |G| > 2 \text{ ou } e = 0 \end{array} \right.$$

Preuve :

i) Cas où $\tau_G(\alpha) \neq \tau_G(\beta)$

on peut par raison de symétrie supposer que $\tau_G(\alpha) < \tau_G(\beta)$

prenons un $u \in L(G,\alpha) \ / \ |u| = \tau_G(\alpha)$

soit $u \notin FG(L(G,\beta))$ donc $Divg(\alpha,\beta) \leqslant |u| = \min(\tau_G(\alpha),\tau_G(\beta))$

soit $u \in FG(L(G,\beta))$

or $u \notin L(G,\beta)$ car $|u| < \tau_G(\beta)$

donc $\exists \ a \in \Sigma \ / \ ua \in FG(L(G,\beta))$

mais $ua \notin FG(L(G,\alpha))$ car G est préfixe et $u \in L(G,\alpha)$

donc $Divg(\alpha,\beta) \leqslant |ua| = \min(\tau_G(\alpha),\tau_G(\beta)) + 1$

la proposition 6.3.2 est donc vraie dans le cas où $\tau_G(\alpha) \neq \tau_G(\beta)$

ii) Cas où $\tau_G(\alpha) = \tau_G(\beta)$, $\alpha \not\equiv_G \beta$, $\min(|\alpha|,|\beta|) = 1$

posons $v = \inf_{\leqslant_{lex}} \{u \in Fl(T) \ / \ T(u) \notin I_{\mathbf{N}^*}\}$

Montrons que $T_A^{\tau_G(T(v))+1}(T(v)) = \emptyset$ (1) ?

notons $(\lambda,\mu) = T(v)$. On a alors les 3 cas suivants

soit $\tau_G(\lambda) \neq \tau_G(\mu)$ d' où (1) d' après i)

soit $\tau_G(\lambda) = \tau_G(\mu)$ et $\min(|\lambda|,|\mu|) = 1$

alors par construction algorithmique une T_A transformation

n' est pas applicable i.e. $T_A(T(v)) = \emptyset$ d' où (1)

soit $\tau_G(\lambda) = \tau_G(\mu)$ et $\min(|\lambda|,|\mu|) > 1$

alors par construction algorithmique une T_B transformation

n' est pas applicable i.e. $T_A^{\tau_G(T(v))}(T(v)) = \emptyset$ d' où (1)

dans tous les cas, (1) est établi

comme $\tau_G(\alpha) = \tau_G(\beta)$ et $\alpha \not\equiv_G \beta$ alors $|G| \geqslant 2$

de plus, on peut considérer que $A \neq \emptyset$ et $e \geqslant 1$ car sinon $Divg(\alpha,\beta) = 1$ d' où la proposition

or d' après le lemme 2 $\forall \ u \in Dom(T)$, $\tau_G(T(u)) \leqslant e + 1$

donc la proposition 6.3.1 donne $\{T(v)\} \gg_G T_A^{eCard(A)+1-\tau_G(T(v))}(\alpha,\beta)$

par conséquent avec (1) et par définition de $\gg_G$, on obtient

$$T_A^{eCard(A)+2}(\alpha,\beta) = \emptyset \quad \text{i.e.} \quad Divg(\alpha,\beta) \leqslant eCard(A) + 2 \quad (2)$$

mais par définition de $\prod(\mathbf{N})$, $Card(A) \leqslant |G| - 1$

il nous reste donc à montrer, pour établir entièrement la proposition 6.3.2 , que

Si $Card(A) = |G|-1 \geqslant 2$ alors $Divg(\alpha,\beta) \leqslant e(|G|-1)+1$ (3) ?

prenons $u = \max\limits_{\leqslant lex} Noeud(A)$. On a alors les 2 cas suivants

soit $v \notin D_u$ donc $\exists\ w \in Noeud(A)-\{u\}\ /\ v \in D_w$

d' où $\{T(v)\} \gg_G^{e(Card(A)-1)+1-\tau_G(T(v))} T_A(\alpha,\beta)$ d' après (I) de 6.3.1

il en résulte avec (1) que $Divg(\alpha,\beta) \leqslant e(Card(A)-1)+2 \leqslant e(|G|-1)+1$

soit $v \in D_u$

or d' après (I) de la proposition 6.3.1 et la caractéristique 1 , on a

$$T_A(T(u)) \gg_G^{e(Card(A)-1)+2-\tau_G(T(u))} T_A(\alpha,\beta) \quad (4)$$

notons A l' élément de Ker(G)-Ker(A)

alors $\forall\ \Phi \in T_A(T(u))$, $S(\Phi,A) \in \{A\}^* \times \{A\}^*$

donc $\forall\ \Phi \in T_A(T(u))$, $S(\Phi,A) \in I_{\mathbf{N}^*}$ ou bien est une paire d' éléments de valuation différente

mais $\forall\ (\lambda,\mu) \in A\ \ \tau_G(\lambda) = \tau_G(\mu)$ donc $T(v) \in T_A(T(u))$

par conséquent , avec (1) et (4) on a

$$Divg(\alpha,\beta) \leqslant e(Card(A)-1) + 3 + \tau_G(T(v)) - \tau_G(T(u))$$

mais $u \in Noeud(A)$ donc $\tau_G(T(u)) \geqslant 1$

de plus $T(v) \in T_A(T(u))$ donc $\tau_G(T(v)) \leqslant e$

soit $\tau_G(T(u)) = 1$ alors $T(u) \in \mathbf{N} \times \mathbf{N}$ et donc $\tau_G(T(v)) \leqslant e-1$

d' où $\tau_G(T(v))-\tau_G(T(u)) \leqslant e-2$

soit $\tau_G(T(u)) \geqslant 2$ alors $\tau_G(T(v))-\tau_G(T(u)) \leqslant e-2$

dans tous les cas , $\tau_G(T(v))-\tau_G(T(u)) \leqslant e-2$ donc $Divg(\alpha,\beta) \leqslant e(|G|-1)+1$

(3) est donc établie

iii) en définitive i) et ii) établissent entièrement la proposition 6.3.2 □

Proposition 6.3.3

Soient $\alpha, \beta \in \mathbf{N}^*\ /\ \tau_G(\alpha) = \tau_G(\beta)$, $\alpha \not\equiv_G \beta$, $\min(|\alpha|,|\beta|) > 1$

alors $Divg(\alpha,\beta) \leqslant \begin{cases} \tau_G(\alpha) + (|G|-1)e & \text{si } |G| > 2 \text{ ou } e = 0 \\ \tau_G(\alpha) + (|G|-1)e + 1 & \text{sinon} \end{cases}$

Preuve : Montrons l' inégalité par induction sur $\min(|\alpha|,|\beta|) \geqslant 1$

avec $\alpha, \beta \in \mathbf{N}^*$, $\tau_G(\alpha) = \tau_G(\beta)$, $\alpha \not\equiv_G \beta$

pour $\min(|\alpha|,|\beta|) = 1$ le résultat est donné par la proposition 6.3.2

supposons l' inégalité vraie pour $\min(|\alpha|,|\beta|) = n$ avec $n \geqslant 1$, $\tau_G(\alpha) = \tau_G(\beta)$ et $\alpha \not\equiv_G \beta$ alors montrons là pour $\min(|\alpha|,|\beta|) = n+1$ avec $\tau_G(\alpha) = \tau_G(\beta)$ et $\alpha \not\equiv_G \beta$

on peut par raison de symétrie supposer que $\tau_G(\alpha(1)) \geqslant \tau_G(\beta(1))$

soit $T_A(\alpha,\beta)^{\tau_G(\beta(1))} = \emptyset$ alors $\mathrm{Divg}(\alpha,\beta) \leqslant \tau_G(\beta(1)) \leqslant \tau_G(\alpha)$

soit $T_A(\alpha,\beta)^{\tau_G(\beta(1))} \neq \emptyset$ donc $T_A(\alpha(1),\beta(1))^{\tau_G(\beta(1))} \neq \emptyset$ et $e \neq 0$

alors une T_B transformation est applicable en (α,β)

i.e. $\exists\ \gamma \in \mathbf{N}^* \ /\ (\gamma,\perp) \in T_A(\alpha(1),\beta(1))^{\tau_G(\beta(1))}$

et $T_B(\alpha,\beta) = \{ (\alpha(1),\beta(1)\gamma), (\gamma(\alpha\backslash 2),\beta\backslash 2) \}$

et d' après la caractéristique 2

$\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\alpha(1),\beta(1)\gamma) + \tau_G(\alpha) - \tau_G(\alpha(1),\beta(1)\gamma)$ (1)

$\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\gamma(\alpha\backslash 2),\beta\backslash 2) + \tau_G(\beta) - \tau_G(\gamma(\alpha\backslash 2),\beta\backslash 2)$ (2)

si $\tau_G(\alpha(1)) \neq \tau_G(\beta(1)\gamma)$

(1) et 6.3.2 $\Rightarrow$ $\mathrm{Divg}(\alpha,\beta) \leqslant \tau_G(\alpha) + 1$

si $\tau_G(\gamma(\alpha\backslash 2)) \neq \tau_G(\beta\backslash 2)$ alors on voit de même avec (2)

si $\tau_G(\alpha(1)) = \tau_G(\beta(1)\gamma)$ et $\tau_G(\gamma(\alpha\backslash 2)) = \tau_G(\beta\backslash 2)$ alors (1) et (2)

deviennent $\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\alpha(1),\beta(1)\gamma)$ (3)

$\mathrm{Divg}(\alpha,\beta) \leqslant \mathrm{Divg}(\gamma(\alpha\backslash 2),\beta\backslash 2) + \tau_G(\beta(1))$ (4)

posons k = Si $|G| > 2$ Alors 1 Sinon 2 ls

soit $\alpha(1) \not\equiv_G \beta(1)\gamma$ alors d' après la proposition 6.3.2 et (3)

$\mathrm{Divg}(\alpha,\beta) \leqslant (|G|-1)e + k + \tau_G(\alpha\backslash 2)$

d' où $\mathrm{Divg}(\alpha,\beta) \leqslant (|G|-1)e + k-1 + \tau_G(\alpha)$

soit $\alpha(1) \equiv_G \beta(1)\gamma$ donc $\gamma(\alpha\backslash 2) \not\equiv_G \beta\backslash 2$ car T_B est valide

l' hypothèse d' induction et (4) donnent

$\mathrm{Divg}(\alpha,\beta) \leqslant (|G|-1)e + k-1 + \tau_G(\beta\backslash 2) + \tau_G(\beta(1))$

i.e. $\mathrm{Divg}(\alpha,\beta) \leqslant (|G|-1)e + k-1 + \tau_G(\alpha)$

dans tous les cas , l' inégalité est satisfaite

l' induction est terminée et la proposition 6.3.3 établie $\square$

Conclusions

De nombreux travaux portent sur l' algorithme de Korenjak-Hopcroft (ou un dérivé) et doivent donc être réanalysés sous un angle nouveau .
De plus , comme déjà indiqué dans l' introduction , cette méthode est généralisable à d' autres algorithmes de décision .
Enfin et surtout , outre l' optimisation d' algorithmes existants , " l' arrêt par conséquence théorique" devrait permettre d' entrebailler la vieille porte de la décidabilité de l' égalité des langages algébriques déterministes .

Remerciements

Je remercie Laurent KOTT de m' avoir soutenu tout au long de ce travail .
Je remercie l' IRISA de m' avoir accueilli .

Références

Bu 73 — P. Butzbach "Sur l' équivalence des grammaires simples" *Actes des premières journées d' informatique théorique . Langages algébriques* . Bonascre . pp 223-245

Co 74 — B. Courcelle "Une forme canonique pour les grammaires simples déterministes" *Rairo* . pp 19-36

Co 83 — B. Courcelle "An axiomatic approach to the KH algorithms" *Math. Systems Theory* . pp 191-231

Fr 77 — E.P. Friedman "Equivalence problems for deterministic context-free languages and monadic recursion schemes" *Journal of Computer and System Sciences"* . pp 344-359

Fr 76 — E.P. Friedman "The inclusion problem for simple languages" *Theoret. Comput. Sci. 1* . pp 297-316

Ha 78 — M.A. Harrison *Introduction to formal language theory* Addison-Wesley

Ha-Ha Ye 79 — M.A. Harrison , I.M. Havel , A. Yeduhaï "On equivalence of grammars through transformation trees" *Theoret. Comput. Sci. 9* . pp 191-231

Ho-Ul 79 — J.E. Hopcroft , J.D. Ullman *Introduction to automata theory , languages ,and computation* . Addison-Wesley

Hu 80 — G. Huet "Confluent reductions : abstract properties and applications to term rewriting systems" *Journal of the Assoc. for Comput. Mach.* . pp 797-821

Ko-Ho 66 — A.J. Korenjak , J.E. Hopcroft "Simple deterministic languages" *Seventh Annual IEEE Switching and Automata Theory Conference* . pp 36-46

Ol-Pn 77 — T. Olshansky , A. Pnueli "A direct algorithm for checking equivalence of LL(k) grammars" *Theoret. Comput. Sci. 4* . pp 321-349

To 83 — E. Tomita "A direct branching algorithm for checking equivalence of strict deterministic vs. LL(k) grammars" *Theoret. Comput. Sci. 23* . pp 129-154

Wo 73 — D. Wood "Some remarks on the KH algorithm for s-grammars" *BIT 13* . pp 476-489

PI 224 **Présentation simplifiée d'une machine de gestion de mémoire pour les interpréteurs Prolog**
Yves Bekkers, Bernard Canet, Ollivier Ridoux, Lucien Ungaro, 20 pages ; Février 1984.

PI 225 **Algorithmes adaptatifs pour l'estimation du décalage entre deux signaux numériques**
Michèle Basseville, Danielle Pelé, 43 pages ; Avril 1984.

PI 226 **Modélisation avec pivot pour une loi générale**
Jean Pellaumail, 11 pages ; Mai 1984.

PI 227 **Systèmes de processus communicants et interprétation parallèle de langages fonctionnels**
Boubakar Gamatié, 30 pages ; Juin 1984.

PI 228 **Un nouveau réseau d'interconnexion adapté aux calculateurs SIMD**
André Seznec, 42 pages ; Juillet 1984.

PI 229 **Analyse factorielle en référence à un modèle. Application à l'analyse de tableaux d'échanges**
Brigitte Escofier, 19 pages ; Juillet 1984.

PI 230 **Infinitary languages and fully abstract models of fair asynchrony**
Philippe Darondeau, 79 pages ; Juillet 1984.

PI 231 **Contribution d'une approche syntaxique dans la segmentation d'image**
Jean Camillerapp, Ivan Leplumey, 46 pages ; Juillet 1984.

PI 232 **Analyse d'un algorithme de classification hiérarchique «en parallèle» pour le traitement de gros ensembles**
Israël-César Lerman, Philippe Peter, 114 pages ; Août 1984.

PI 233 **Manuel d'utilisation de Diastol - Version Préliminaire**
Patrice Quinton, Pierrick Gachet, 29 pages ; Août 1984.

PI 234 **Architectures systoliques pour la reconnaissance de mots connectés (en anglais)**
François Charot, Patrice Frison, Patrice Quinton, 40 pages ; Août 1984.

PI 235 **A scheme of Token Tracker**
Zhao Jing Lu, 62 pages ; Septembre 1984.

PI 236 **Design of one-step and multistep adaptive algorithms for the tracking of time varying systems**
Albert Benveniste, 40 pages ; Septembre 1984.

PI 237 **The design and building of Enchère, a distributed electronic marketing system**
Jean-Pierre Banatre, Michel Banatre, Guy Lapalme, Florimond Ployette, 38 pages ; Septembre 1984.

PI 238 **Algorithme optimal de décision pour l'équivalence des grammaires simples**
Didier Caucal, 48 pages ; Septembre 1984.

ALGORITHME OPTIMAL DE DECISION POUR L'EQUIVALENCE DES GRAMMAIRES SIMPLES

Didier CAUCAL

Publication n° 238

Septembre 1984